

## Example-guided image editing

Hristova Hristina

► To cite this version:

Hristova Hristina. Example-guided image editing. Engineering Sciences [physics]. Université de Rennes 1 [UR1], 2017. English. tel-01737486

**HAL Id: tel-01737486**

**<https://hal.inria.fr/tel-01737486>**

Submitted on 19 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THÈSE / UNIVERSITÉ DE RENNES 1**  
*sous le sceau de l'Université Européenne de Bretagne*

pour le grade de  
**DOCTEUR DE L'UNIVERSITÉ DE RENNES 1**

*Mention : Informatique*  
**Ecole doctorale MATISSE**

présentée par  
**Hristina HRISTOVA**

préparée à l'unité de recherche UMR 6074 IRISA  
et au centre IRISA - Rennes Bretagne Atlantique  
ISTIC

**Example-guided  
image editing**

**Thèse soutenue à 20 Octobre 2017 à Rennes**  
devant le jury composé de :

**Joëlle THOLLOT**

Professeur à Univ. de Grenoble / rapporteur

**Alan CHALMERS**

Professor à Univ. de Warwick / rapporteur

**Marcelo BERTALMÍO**

Associate professor à Univ. Pompeu Fabra / examinateur

**Nicolas BONNEEL**

Chargé de Chercheur CNRS, LIRIS / examinateur

**Eric MARCHAND**

Professeur ESIR / examinateur

**Kadi BOUATOUCH**

Professeur à Univ. de Rennes 1 / examinateur

**Rémi COZOT**

Maître de Conférences à Univ. de Rennes 1 / directeur de thèse

**Olivier LE MEUR**

Maître de Conférences à Univ. de Rennes 1 / co-directeur de thèse





# Abstract

This thesis addresses three main topics from the domain of image processing, i.e. color transfer, high-dynamic-range (HDR) imaging and guidance-based image filtering. The first part of this thesis is dedicated to color transfer between input and target images. Color transfer is often viewed as a distribution transfer problem in which the image color distributions are modelled using the multivariate Gaussian distribution (MGD). Existing color transformations rely on the accuracy of the MGD model and may fail to produce plausible results when the MGD does not fit well enough the distribution of the image colors. To overcome this limitation, in this thesis we adopt cluster-based techniques. We apply Gaussian mixture models to partition the input and target images into Gaussian clusters (each cluster follows an MGD). In addition, we propose four new mapping policies to efficiently map the target clusters to the input clusters. Our results and evaluation show a significant improvement over existing color transfer methods.

Color transfer is limited to transferring color between images. To address this limitation, we exploit the properties of the multivariate generalized Gaussian distributions (MGGD). The MGGD can fit a wide class of image features distributions, including the distributions of color, gradient, wavelet coefficients, etc. We propose a novel transformation of the MGGD, which we apply to simultaneously transfer both color and gradient. The proposed MGGD transformation proves to be beneficial to other image processing tasks, such as color correction.

Even though the MGGD and the MGD are both continuous distributions, they are often adopted to model the discrete distributions of color and light in images. Our experiments have shown that the bounded Beta distribution provides a much more precise model for the color and light distributions of images. To exploit this property of the Beta distribution, we propose a new color transfer method, where we model the color and light distributions by the Beta distribution. To this end, we introduce a novel transformation of the Beta distribution. The results, obtained by applying our Beta transformation, appear more natural and less saturated than results from recent state-of-the-art methods. Additionally, our results represent accurately the target color palette and truthfully portray the target contrast.

Different color transfer methods often result in different output images. The process of determining the most plausible output image may be subjective, as it depends on a person's preference. To lessen the level of subjectivity in quality assessment for color transfer, in this thesis we propose a model for objective evaluation of the color transfer. Our model explains the relationship between users' perception and a number of perceptual image features.

The second part of this thesis focuses on HDR imaging. First, we present a color transfer method between HDR images. To this end, we propose an extension of existing color transfer methods to the HDR domain. Second, we introduce a method for automatic

creation of HDR images from only two images - flash and non-flash images. We mimic the camera response function by a brightness function to obtain a number of differently exposed images using only the non-flash image. Then, we recover details from the flash image using our new chromatic adaptation transform (CAT), called bi-local CAT. That way, we efficiently recover the dynamic range of the real-world scene without compromising the quality of the HDR image (as our method is robust to misalignment).

In the context of the HDR image creation, the bi-local CAT recovers details from the flash image, removes flash shadows and reflections. In the last part of this thesis, we exploit the potential of the bi-local CAT for various image editing applications such as image de-noising, image de-blurring, texture transfer, etc. We propose a novel guidance-based filter in which we embed the bi-local CAT. The proposed filter performs as good as (and for certain applications even better than) state-of-the art methods.

# Acknowledgements

First, I would like to thank my two supervisors: Olivier Le Meur and Rémi Cozot. I would like to thank Olivier for motivating me to push beyond my limits and to evolve, and Rémi for all the fruitful discussions and ideas. Their doors were always open and they were always willing to help me. I learned a lot about research from each one of them!

I would like to specially thank my third (though not official) supervisor Kadi Bouatouch for his time, support and attention. His guidance was very valuable to me!

I would like to thank Joëlle Thollot, Alan Chalmers, Marcelo Bertalmío, Erik Reinhard, Nicolas Bonneel and Eric Marchand for having accepted to be a part of my Ph.D. defense committee.

Many thanks to all the current and formal members of the FRVSense team for creating a positive and friendly working atmosphere. Special thanks to Adrien for the friendly pieces of advice and help.

I would also like to thank all my close relatives and friends for all the moments of joy and laughter we shared.

Finally, I would like to express my gratitude to my parents and my sister for their unfailing love and support. I specially thank my boyfriend, Ivo, for his unconditional love and continuous encouragement throughout these three years. This thesis would not have been possible without them. Thank you!

# Contents

<b>List of figures</b>	<b>11</b>
<b>List of tables</b>	<b>15</b>
<b>1 Introduction</b>	<b>17</b>
1 Color transfer . . . . .	17
1.1 Distribution models . . . . .	18
1.2 Quality assessment . . . . .	19
2 HDR imaging . . . . .	19
3 Guidance-based image filtering . . . . .	20
4 Summary of contributions . . . . .	20
5 Organization of the thesis . . . . .	21
6 List of publications . . . . .	22
 <b>I Background</b>	 <b>23</b>
<b>2 Colorimetry and color spaces</b>	<b>27</b>
1 Colorimetry . . . . .	27
1.1 Brightness . . . . .	27
1.2 Lightness . . . . .	27
1.3 Hue . . . . .	27
1.4 Color saturation . . . . .	27
2 Color spaces . . . . .	28
2.1 RGB . . . . .	28
2.2 CIE XYZ . . . . .	28
2.3 LMS . . . . .	29
2.4 HSV . . . . .	29
2.5 CIE Lab . . . . .	29
2.6 Lch . . . . .	30
 <b>3 Chromatic adaptation transform and white point</b>	 <b>31</b>
1 White point . . . . .	31
2 Chromatic adaptation transform . . . . .	31
 <b>4 Distribution models</b>	 <b>35</b>
1 Multivariate generalized Gaussian distribution . . . . .	35
2 Multivariate Gaussian distribution . . . . .	36

3	Multivariate Laplace distribution . . . . .	36
4	Beta distribution . . . . .	37
<b>II Distribution-based transfer for example-guided image editing</b>		<b>39</b>
<b>5</b>	<b>Style-aware robust color transfer</b>	<b>43</b>
1	Introduction . . . . .	43
2	Related work . . . . .	44
2.1	Color transfer . . . . .	44
2.1.1	Non-parametric methods . . . . .	45
2.1.2	Parametric methods . . . . .	45
2.1.3	Limitations . . . . .	46
2.2	Style transfer . . . . .	46
2.2.1	Non-photo-realistic style transfer . . . . .	47
2.2.2	Photo-realistic style transfer . . . . .	48
3	Style-aware style transfer . . . . .	48
3.1	Automatic light-based versus colors-based classification . . . . .	49
3.2	Clustering . . . . .	51
3.3	Mapping policies . . . . .	51
3.3.1	Light to Colors . . . . .	52
3.3.2	Colors to Light . . . . .	53
3.3.3	Light to Light . . . . .	53
3.3.4	Colors to Colors . . . . .	54
3.4	Color transfer method . . . . .	55
3.4.1	Color transformation on the chroma channels . . . . .	55
3.4.2	Overlapping . . . . .	55
3.4.3	Local chromatic adaptation . . . . .	56
3.5	Implementation details . . . . .	57
4	Results and evaluation . . . . .	57
4.1	User study evaluation . . . . .	59
4.2	Objective metric evaluation . . . . .	61
5	Limitations . . . . .	63
6	Conclusion . . . . .	63
<b>6</b>	<b>Transformation of the Multivariate Generalized Gaussian Distribution for Image Editing</b>	<b>65</b>
1	Introduction . . . . .	65
2	Related work on MGD transformations . . . . .	66
3	Transformation of the MGGD . . . . .	67
3.1	Multivariate generalized Gaussian distribution . . . . .	67
3.2	Transformation between two MGGDs . . . . .	69
3.2.1	Monge-Kantorovich closed-form mapping . . . . .	69
3.2.2	Stochastic-based shape parameter transfer . . . . .	70
3.2.3	The final transformation $\mathbf{T}_{\mathbf{u} \rightarrow \mathbf{v}}$ . . . . .	72
3.2.4	Properties of the transformation $\mathbf{T}_{\mathbf{u} \rightarrow \mathbf{v}}$ . . . . .	72
3.2.5	Evaluation . . . . .	73

4	Applications	74
4.1	Gradient transfer	75
4.2	Color transfer (3D)	79
4.3	Simultaneous color and gradient transfer (5D)	80
4.4	Color correction	82
5	Conclusion and future work	83
<b>7</b>	<b>Beta distribution transformation for color transfer</b>	<b>85</b>
1	Introduction	85
2	Beta transformation	86
2.1	Beta distribution	86
2.1.1	Beta-Fisher relationship	86
2.1.2	Fisher-Chi-square relationship	87
2.2	Transformation between Beta distributions	87
2.2.1	Beta-to-Fisher	88
2.2.2	Fisher-to-Standard-Normal	88
2.2.3	Standard-Normal-to-Fisher	89
2.2.4	Fisher-to-Beta	89
2.2.5	Choice of p	90
2.3	Evaluation of Beta transformation	90
3	Results	90
3.1	Global color transfer	91
3.2	Local color transfer	91
4	Conclusion	93
<b>8</b>	<b>Perceptual metric for color transfer methods</b>	<b>97</b>
1	Introduction	97
2	Related works	98
3	Our method	99
3.1	User study	100
3.2	The regression model	100
4	Results	102
5	Conclusion	104
<b>III</b>	<b>Example-guided high-dynamic-range image editing</b>	<b>107</b>
<b>9</b>	<b>Color transfer between high-dynamic-range images</b>	<b>111</b>
1	Introduction	111
2	Why do LDR color transfer methods need to be extended to HDR images?	111
3	Adapting a color transfer method to HDR images	112
3.1	Style-aware color transfer [7]	112
3.2	Extension to the HDR domain	113
3.2.1	Color space conversion	114
3.2.2	Image classification	114
3.2.3	Clustering and mapping	114

3.2.4	Cluster-based local CAT	115
3.3	Results	117
3.3.1	Protocol	117
3.3.2	Objective evaluation	117
3.4	Is the proposed extension applicable to state-of-the-art color transfer methods?	119
4	Conclusion	121
<b>10 High-Dynamic-Range Image Recovery from Flash and Non-Flash Image Pairs</b>		<b>123</b>
1	Introduction	123
2	Related works	124
3	Our method	126
3.1	Noise removal	126
3.2	Brightness gamma correction	126
3.3	Iterative bi-local CAT	127
3.4	Image fusion	130
3.5	Choice of optimal values of $p$ and $N$	130
4	Results and evaluation	132
5	Conclusion	137
<b>IV Perceptual-guided filtering for image enhancement and image editing</b>		<b>139</b>
<b>11 Multi-purpose CAT-based guidance filter</b>		<b>143</b>
1	Introduction	143
2	Related work	144
2.1	Convolution-based guidance filters	144
2.1.1	Cross-bilateral filter	145
2.1.2	Rolling guidance filter	145
2.2	Model-based guidance filters	145
2.2.1	Guided filter	146
2.2.2	Weighted guided filter	148
2.2.3	Distribution-based transformations	148
2.3	Discussion	149
3	Our method	149
3.1	Bi-local CAT in the frame of guidance-based filters	150
3.2	CAT-based guidance filter	151
4	Implementation	154
5	Results	154
6	Conclusion	163
<b>12 Conclusion</b>		<b>165</b>
1	Conclusion	165
2	Future work	166
2.1	Example-guided color transfer	166



## CONTENTS

---

2.2	Multivariate Beta distribution . . . . .	167
2.3	Fisher distribution . . . . .	167
2.4	Video color transfer . . . . .	168
2.5	Style transfer . . . . .	168
2.6	360-degrees images . . . . .	169
2.7	HDR image creation . . . . .	169
2.8	HDR video sequences . . . . .	170
<b>13 Appendices</b>		<b>171</b>
<b>APPENDICES</b>		<b>171</b>
A	Demonstration of the choice of $\hat{\tau}_{\mathbf{w}_i}$ and $\hat{\mathbf{e}}_{\mathbf{w}_i}$ in equations 6.12 and 6.13 . . . . .	171
B	Proof of Proposition 3.2.1 . . . . .	172

# List of Figures

2.1	Hue and saturation. . . . .	28
3.1	Color appearance - influence of the surrounding factor. . . . .	32
4.1	Comparison between MGD, MLD and Beta distribution. . . . .	37
5.1	Illustration of one of the limitations of the Gaussian assumption for color transformations. . . . .	44
5.2	Artistic style transfer. . . . .	47
5.3	Main flowchart. . . . .	49
5.4	Classification system - parameter adjustments. . . . .	50
5.5	Example of images correctly and incorrectly classified by our classification system. . . . .	51
5.6	Illustration of clustering according to the luminance histogram and the 2D Luminance-Hue joint distribution. . . . .	52
5.7	Illustration of the photographic approach using warm and cold colors to indicate highlights and shadows in images. . . . .	54
5.8	Comparison between the naive histogram matching, the global CAT and the local CAT. . . . .	57
5.9	Results from our method and four state-of-the-art methods. . . . .	58
5.10	Screen shot of the system we designed to conduct our user study. . . . .	60
5.11	Subjective evaluation - analysis using Box-and-Whisker plots. . . . .	60
5.12	Objective evaluation - analysis using contour plots for the metric pair (SSIM, Bhattacharya). . . . .	62
5.13	Additional results. . . . .	64
6.1	Comparison between the MGD, the MLD and the MGGD for $p = 1$ (univariate case). . . . .	67
6.2	Main flowchart. . . . .	68
6.3	Box-and-Whisker plots of the percentage error of the MGGD proposed transformation. . . . .	73
6.4	Illustration of the continuity and the non-linearity of the proposed MGGD transformation. . . . .	75
6.5	Results from a gradient transfer between images. . . . .	77
6.6	Results from a color transfer between images, obtained with our 3D MGGD-based method and Pitié's and Reinhard's MGD-based methods. . . . .	79

6.7	Result from a simultaneous 5D transfer of color and gradient, compared to a result from a 3D color transfer (obtained with our transformation). . . . .	81
6.8	Color correction results. . . . .	82
6.9	Comparison between results of a simultaneous 5D MGGD-based transfer of color and gradient, 5D MGD-based transfer of color and gradient and 3D MGGD-based transfer of color. . . . .	84
7.1	Gaussian and Beta approximations of the lightness in images. . . .	86
7.2	Main flowchart. . . . .	87
7.3	Box-and-Whisker plots of the percentage errors for the two Beta shape parameters (computed after 1 and 5 iterations of our transformation). . . . .	90
7.4	Results of global color transfer. . . . .	91
7.5	Results of our Beta transformation and comparison with state-of-the-art color transfer methods. . . . .	92
7.6	Lightness-Hue distribution plot for comparing results of color transfer. .	93
7.7	Results of our Beta transformation (local and global) and comparison with state-of-the-art color transfer methods. . . . .	94
7.8	Additional results. . . . .	95
8.1	Objective evaluation of color transfer results. . . . .	98
8.2	Main flowchart. . . . .	99
8.3	Box-and-Whisker plots for the results of our user study and the predictions, obtained using linear, non-linear regressions and random forests. . . . .	99
8.4	Saliency and color transfer. . . . .	101
8.5	Relationship between the predicted scores from our model and the actual scores from our user study. Residual distribution in our model. .	102
8.6	Bar plots of the importance of each objective feature in our model. .	103
8.7	Score predictions from random forests, linear and non-linear regressions. . . . .	104
9.1	The main flowchart of our style-aware color transfer method [7] including the proposed extension. . . . .	113
9.2	Light-based and colors-based style images. . . . .	113
9.3	An HDR image and its three regions, corresponding to the peaks of its log-luminance histogram. . . . .	116
9.4	Results from applying our cluster-based CAT algorithm. . . . .	117
9.5	Results obtained by applying our style-aware color transfer method [7] (without any modifications) and its extension to HDR images. . . . .	118
9.6	Box-and-Whisker plots of both SSIM and Bhattacharya coefficient for results obtained using our style-aware color transfer method (without modifications) and its HDR extension. . . . .	119
9.7	Results, obtained by applying state-of-the-art color transfer methods (directly, without any modifications), and results, obtained with their extensions to HDR images. . . . .	120

10.1	Main flowchart. . . . .	125
10.2	Comparison between bi-local CAT and local CAT. . . . .	127
10.3	Comparison between guided filter, bilateral filter and Gaussian filter, used to compute the white images in our method. . . . .	129
10.4	Illustration of the darkening and brightening properties of the bi-local CAT and the influence of the parameter $p$ . . . . .	130
10.5	CRF computation and comparison with the ground truth. . . . .	131
10.6	Plots of the SSIM scores between a real multi-exposure image and each of 400 multi-exposure images, obtained with our method. . . . .	132
10.7	HDR result, obtained with our method, and comparison with the ground truth. . . . .	133
10.8	Comparison between our method and classical multi-exposure methods. . . . .	134
10.9	Result of applying our method to dark environment scenes with high dynamic range. Detail recovery. . . . .	134
10.10	Result of applying our method to portraits. . . . .	135
10.11	Flash shadows and reflections removal. . . . .	135
10.12	Non-flash image detail recovery and enhancement. . . . .	136
10.13	Non-flash image enhancement. . . . .	136
10.14	Limitation. . . . .	136
11.1	Results of applying guided filter in the context of image denoising. . . . .	147
11.2	Main flowchart. . . . .	151
11.3	Influency of the white image ratio on the detail recovery from the guidance image. . . . .	153
11.4	Influence of the two parameters of the CAT-based filter. . . . .	155
11.5	Results of image denoising. . . . .	157
11.6	Influence of the iterations of the CAT-based filter. . . . .	157
11.7	Results of texture transfer. . . . .	158
11.8	Results of detail enhancement with an NIR image. . . . .	159
11.9	Results of iterative image deblurring. . . . .	160
11.10	Results of mask refinement. . . . .	161
11.11	Results of skin beautification with NIR image. . . . .	161
11.12	Results of single-image skin beautification. . . . .	162
A1	The distributions of the $\lambda^{th}$ power ( $\lambda = 1/4$ ) of samples, drawn from gamma distributions, and their corresponding normal approximations. . . . .	172
B1	Gamma distributions with varying shape parameters and fixed scale parameters. . . . .	173



# List of Tables

5.1	Correlations between the style and aesthetics scores. . . . .	48
5.2	Correlations between the style and aesthetics scores. . . . .	62
5.3	Mean values of the SSIM and the Bhattacharya coefficient for each of five methods . . . . .	62
8.1	Correlation $\pm$ standard deviation and MSE over all 10 test sets in our cross-validation. . . . .	103
9.1	General modifications for adapting color transfer methods to HDR images. . . . .	120
11.1	Notations. . . . .	144



# Introduction

Each day millions of images are being uploaded to social medias, such as Instagram, Google and Facebook. These platforms allow users to stylize their images by applying various filters to them or by adjusting the levels of brightness, contrast, colorfulness, etc. A major part of all uploaded images is taken with a mobile camera and is thus prone to noise and blur. Noise and blur removal tools become essential for enhancing the appearance of the user photos. The popularity of the social media platforms has directed the research interest towards innovative, more advanced image editing algorithms for improving the quality of images and mainly, for image stylization.

The works, presented in this thesis, contribute to the domain of example-guided image editing. The topics, which we address, are three-fold, *i.e.* image color transfer, high-dynamic-range (HDR) imaging and guidance-based image filtering. They are introduced in the following sections.

## 1 Color transfer

The primary focus of this thesis is the image color transfer. Color transfer refers to the process of modifying the colors of an input image so that they match the target color palette. Usually, color transfer is addressed as a problem of transforming statistical distributions. To this end, a color space representation of the input and target color pixels is computed. The distributions of the input and target pixels in the chosen color space are used to compute a transformation between the colors of the input and target images. Given this transformation, the input color distribution can be transformed into a distribution that is similar to the target color distribution. For simplicity sake, color transformations are computed by assuming that the input and target color distributions follow a certain distribution model. The quality of the color transfer strongly depends on how well the chosen distribution model fits the input and target color distributions. In this thesis, we study the potential of applying various distribution models in the context of color transfer. In this sense, this thesis answers the following research questions: *Which distribution models and color mappings improve the quality of classical color transfer methods? How is the transfer of color between images generalized to a multidimensional transfer of image features? How is the quality of a color transfer objectively assessed?*

To address the aforementioned questions, we benefit from the properties of several well-known distribution models, used in image processing. Hereafter, we present these models as well as their advantages and limitations when applied in the context of color transfer.



## 1.1 Distribution models

The multivariate Gaussian distribution (MGD) is one of the most important and well-studied continuous distributions. The MGD has an analytically tractable density function and therefore, it is beneficial to many applications, such as example-based color transfer [1], color correction for image mosaicking [2], etc. The MGD has been commonly used to model the distributions of color and light in images [3]. However, a single MGD does not always fit well enough the color and light distributions. This becomes evident in cases when the image is composed of more than one color or when the image contains two luminance clusters, *i.e.* highlights and shadows. The Gaussian model can be extended to Gaussian mixture models (GMMs). GMMs describe a given distribution as a mixture of several MGDs. In general, GMMs provide a decent model for many complex distributions. However, the accuracy of GMMs depends on the number of mixtures, which is given as an input (*i.e.* the number of mixtures is not determined by GMMs but it needs to be computed in advance). In addition, like the MGD, GMMs fail to describe well heavy-tailed distributions.

Heavy-tailed distributions, such as the distributions of image gradient and wavelet coefficients, are often modelled using the multivariate Laplace distribution (MLD). The MLD and the MGD belong to the same class of distributions, called the multivariate generalized Gaussian distributions (MGGD). The MGGD with varying shape parameter encompasses the majority of elliptical heavy-tailed and light-tailed distributions. To this end, it can fit the distributions of a wide class of image features, *e.g.* color, light, gradient, wavelet coefficients, etc. In contrast, the MGGD (and its special cases the MGD and the MLD) fits only symmetrical distributions and it does not account for the skewness of the modelled distributions.

The distributions of image features, such as color and light, are bounded in a finite interval. Despite that, continuous distributions (mostly the MGD) are widely used to describe these discrete distributions. There exists a number of bounded distributions, such as the Beta distribution, the Fisher distribution, etc. These bounded distributions are asymmetrical and describe well the skewness of a given discrete distribution. Recent research has shown the benefit of modelling an unknown discrete distribution using bounded distributions [4].

Some of the distribution models have been utilized in the context of color transfer between input and target images. Early research methods compute global color mappings between the input and target images upon the assumption that the color and light distributions of both images can be fitted by the MGD [3]. In general, the Gaussian assumption turns out to be too restrictive to ensure a good color transfer. To improve the results of global methods, image clustering has been adopted [5,6]. Local methods partition the distributions of the input and target images into clusters (usually using GMMs) and perform the color mappings locally between corresponding clusters. Before performing the color transfer, the input and target clusters are first mapped one to another using a mapping function, based on the light distributions of the clusters. So far, the mapping function between the clusters has been carried out regardless of the color distributions of the clusters.

As discussed earlier in this section, the MGGD presents a general model for describing a number of image features, including color and gradient. Therefore, the MGGD could

be efficient for performing a multi-feature transfer (for instance, color and gradient joint transfer) between two images. However, the potential of the MGGD in this context has not yet been exploited. Likewise, the bounded distributions have not yet played a role in computing color transformations, even though their enormous potential for modelling image feature distributions.

---

## 1.2 Quality assessment

A different color transformation needs to be derived for each given distribution model. Furthermore, different color transformations produce different results. To objectively assess the performance of color transformations, objective metrics (independently or as a combination) could be used [7]. However, a standalone objective evaluation is often insufficient. The process of assessing the quality of a color transfer and evaluating the plausibility of the final result is subjective and is influenced by a person's preference. To compare results of color transformations one to another, user studies are commonly conducted. Conducting a user study every time a transformation is proposed is a tedious task. Yet, it is widely used due to a lack of a perceptual metric, measuring the quality of a color transfer. This thesis proposes a novel solution for easing the quality assessment of color transfer methods.

---

# 2 HDR imaging

In the first part of this thesis, we tackle a color transfer between low-dynamic-range (LDR) images, *i.e.* digital images whose values lie in the displayable range. The second part of the thesis addresses the problem of transferring color between high-dynamic-range (HDR) images. HDR images portray the high luminance of real-world scenes. The luminance range of a single HDR image may vary from extreme dark (*e.g.* starlight) to direct sunlight. The HDR images contain fine details of the original scene both in the highlights and in the shadows. To map the range of the HDR images to the displayable range, *i.e.* to visualize the HDR image on an LDR device, compression algorithms, called tone-mapping operators [8, 9], are applied. The tone-mapping operators aim to approximate the appearance of the HDR images by preserving as many scene details as possible. In spite of that, tone-mapping an HDR image may cause compression artifacts and loss of details and photo-realism. Therefore, the color transfer between HDR images needs to be applied directly in the HDR domain.

The HDR images are commonly created using standard digital cameras. Multi-exposure images, *i.e.* images, taken at various exposure times, are merged together to recover the dynamic range of the HDR images. The more multi-exposure images are merged, the more details are recovered, *i.e.* the closer the HDR image is to the ground truth. The multi-exposure images need to be perfectly aligned to allow an artifact-free recovery of the original scene. To avoid misalignment during the shooting process, photographers are required to use a tripod. Moreover, photographers should choose the correct exposure times in order to capture the finest details of the scene. Challenging cases include scenes containing a moving object and dark-environment scenes. Indeed, a movement in the scene causes the appearance of ghosting artifacts in the final HDR image. Such artifacts

need to be removed. Furthermore, when taking multi-exposures images in dark environment scenes with high-dynamics, one needs to decrease significantly the exposure time in order to capture the details, contained in the highlights. This way, the levels of noise in the shadows augment, which affects the integrity of the final HDR image.

To ease the process of creating HDR images and to tackle the limitations of multi-exposure methods, this thesis proposes an innovative method for recovering an HDR image from only two images. Our method recovers the entire dynamic range of a real-world scene (*i.e.* creates an HDR image) using flash and non-flash image pairs.

---

### 3 Guidance-based image filtering

Flash and non-flash image pairs are widely used as inputs to image filters. Lately, guidance-based filters have raised a lot of research interest thanks to their efficiency in applications such as image de-noising, image de-blurring, detail enhancement. Guidance-based filters process an input image (*e.g.* a non-flash image) by incorporating additional information into the filtering process through the use of a guidance image (*e.g.* a flash image). The guidance image is a sharp noise-free image, which does not represent the ambience of the original scene. Incorporating the guidance image into the estimation of the filter kernel proves to be robust for applications, such as noise removal and image de-blurring, but it may compromise the lighting atmosphere of the input image. For instance, when applied to image de-noising, guidance-based filters exhibit a trade-off between the successful noise removal and the ambience of the original scene. Furthermore, most guidance-based filters are dedicated to image de-noising and are not suited for applications, such as example-guided texture transfer. This thesis addresses the main limitations of guidance-based filters and focuses on designing a new filter with wide range of applications, including image de-noising, texture transfer, skin beautification, etc.

---

### 4 Summary of contributions

The works, presented in this thesis, make the following contributions to the field of image processing:

- a new local method for style-aware color transfer, introducing four novel mapping policies;
- a novel transformation of the MGGD for simultaneous transfer of color and gradient;
- a novel transformation of the Beta distribution for color transfer;
- a perceptual metric for assessing the quality of color transfer methods;
- an extension of the color transfer methods to the HDR domain;
- a new method for creating HDR images using only two LDR images, *i.e.* flash and non-flash images;

- a new guidance-based filter, based on color appearance, applied to example-guided image de-noising, image de-blurring, detail enhancement, texture transfer, etc.

---

## 5 Organization of the thesis

The thesis is divided into four main parts as follows:

- **Part I - Background:** In this part, we introduce the main theoretical concepts used in the rest of the thesis. We present the most relevant (to our work) color spaces, the algorithm for color adaptation, called chromatic adaptation transform (CAT) as well as the most popular distribution models in image processing.
- **Part II - Distribution-based transfer for image editing:** In chapter 5, we introduce a new method, which performs a cluster-based color transfer between input and target images. Our method aims to respect the style of both images by using four novel mapping policies for mapping the input and target clusters. In chapter 6, we extend color transfer to a simultaneous transfer of color and gradient. To this end, we propose a novel transformation of the MGGD with applications to image processing, *e.g.* multidimensional transfer of image features, color correction, etc. Despite the robustness of the MGGD model for applications, such as color and gradient transfers, the MGGD fails to accurately fit asymmetrical distributions. In chapter 7, we adopt a more precise distribution model to fit the distributions of color and light in images, *i.e.* the bounded Beta distribution. We propose a transformation of the Beta distribution which carries out a color transfer between images and outperforms classical Gaussian-based color transformations. To evaluate the quality of the proposed color transfer methods, we finally introduce the first perceptual metric linking the user's subjective evaluation with a set of color appearance attributes (chapter 8).
- **Part III - Example-guided high-dynamic-range image editing:** In chapter 9, we extend the state-of-the-art LDR color transfer methods to the HDR domain by introducing series of adaptation techniques. We demonstrate that the extension algorithm performs better than directly applying a color transfer between HDR images. In chapter 10, we introduce a new method for creating HDR images from flash and non-flash image pairs. Our method consists of a brightness function mimicking the camera response function and a novel CAT, called bi-local CAT, recovering details from the flash image.
- **Part IV - Perceptual guided filtering for image enhancement and editing:** We exploit the potential of our bi-local CAT for various image editing applications such as image de-noising, image de-blurring, texture transfer, etc. We introduce our new guidance-based filter, in which we embed the bi-local CAT, in chapter 11.

## 6 List of publications

- **H. Hristova**, O. Le Meur, R. Cozot, and K. Bouatouch, “Style-aware robust color transfer”, in Proceedings of the workshop on Computational Aesthetics. Eurographics Association, 2015, pp. 67-77.
- **H. Hristova**, O. Le Meur, R. Cozot, and K. Bouatouch, “Color transfer between high-dynamic-range images.” SPIE Optical Engineering Applications. International Society for Optics and Photonics, 2015.
- **H. Hristova**, O. Le Meur, R. Cozot, and K. Bouatouch, “High-dynamic-range image recovery from flash and non-flash image pairs.” *The Visual Computer: International Journal of Computer Graphics* 33.6-8 (2017): 725-735. *Won Best Paper Award at Computer Graphics International 2017, Yohohama, Japan.*
- **H. Hristova**, O. Le Meur, R. Cozot, and K. Bouatouch, “Perceptual metric for color transfer methods”, in IEEE International Conference on Image Processing, 2017.

# **Part I**

## **Background**



# Introduction

This part focuses on concepts from the theory of colorimetry, introduces fundamental color spaces and discusses distribution models, used in image processing to fit various image features. The concepts, introduced in this part, are then utilized in the following parts of the thesis.





# Colorimetry and color spaces

# 2

In this chapter, we introduce fundamental notions from the theory of colorimetry [10] as well as several fundamental color spaces.

---

## 1 Colorimetry

Color is an attribute of the visual perception and it is entirely defined in term of human perception [11]. Color sensations are associated with the light wavelengths reflected or emitted by objects. The cone cells in the human eye are stimulated by the different wavelengths in the spectrum of light. To this end, color sensations distinguish between the different light instances. Color is numerically described as a mixture of three components, called tristimuli, *e.g.* RGB, XYZ, Lab, etc.

---

### 1.1 Brightness

Brightness refers to the visual sensation of a light source, under which an area appears to emit more or less light [10]. The brightness is an absolute color appearance attribute.

---

### 1.2 Lightness

Lightness refers to the relative brightness, normalized by the brightness of the white object in the scene. This means that when the brightness of the scene illuminant changes, the lightness of the white object in the scene remains relatively constant [12].

---

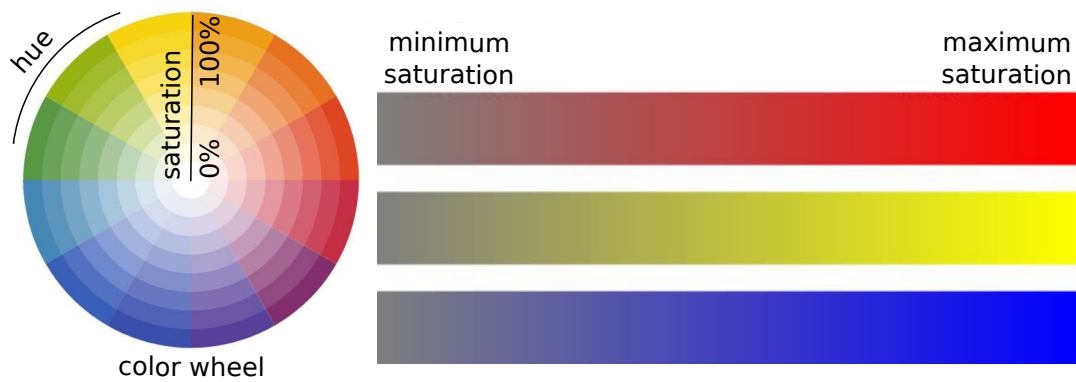
### 1.3 Hue

Hue is an essential color appearance property which defines the resemblance of a color to a pure color (*i.e.* red, orange, yellow, green, blue, violet). The different tints and shades of a given color have the same hue. The color hue is an element of the color wheel [13] and indicates the position of the corresponding pure color (in angles), as shown in figure 2.1.

---

### 1.4 Color saturation

Color saturation describes the intensity of a color in an image. In comparison with the hue, which defines the color itself, the saturation defines the level of purity or vividness of the color. A color is considered pure when it is fully saturated. The saturation varies from pure color (*i.e.* 100%) to gray (*i.e.* 0%) for a constant brightness level. The change in color, caused by varying the saturation, is illustrated in figure 2.1.



**Figure 2.1** – The left-hand side of the figure shows the color wheel. The outermost circle of the wheel contains the pure colors (*i.e.* with the 100% saturation). Three colors, varying with the change of the saturation, are shown on the right-hand side.

---

## 2 Color spaces

According to the trichromatic theory of color vision [14], there are three receptors in the human retina, sensitive to short wavelengths (blue color), medium wavelengths (green color) and long wavelengths (red color). The trichromatic theory states that each color in the visible spectrum can be defined using a combination of these three types of wavelengths. Therefore, each perceived color can be quantified by triplets of numerical values, corresponding to the contribution of the different types of wavelengths. This abstract mathematical model is known as color space representation. There exists a number of color spaces. Hereafter, we discuss the fundamental color spaces, relevant to this thesis.

---

### 2.1 RGB

RGB color space represents each color as a combination of three colors, called tristimulus, *i.e.* red (R), green (G) and blue (B). RGB is most commonly used in digital imaging, video cameras, computer monitors. The three color channels R, G and B are strongly correlated, *i.e.* any modification of a channel may lead to an undesired change of the rest of the channels. To represent any visible color (associated with any wavelength), the red component of the RGB color space model sometimes becomes negative. To address this problem, a new color space, called CIE XYZ, is defined. CIE XYZ can produce every perceived color with positive tristimulus values.

---

### 2.2 CIE XYZ

CIE XYZ is a color space which describes the color perception by three parameters corresponding to short, medium and large level of stimuli of the cone cells in the human eye. The Y channel corresponds to the relative luminance, whereas X and Y depict the response of the human eye to light wavelengths of varying frequencies. CIE XYZ is a device-independent color space as it describes the whole range of color sensations which a person can experience. CIE XYZ is often referred to as absolute, as each primary color

in CIE XYZ is constant. The three primary colors in CIE XYZ are not real colors, they cannot be generated with any light spectrum.

CIE XYZ is obtained as a linear extrapolation of RGB to avoid negative values. The linear transformation from RGB to CIE XYZ takes into account the white point of the display. The most common conversion matrix (and the one, used in this thesis) is given in the following equation (for CIE D65 white point [15, 16]):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3575 & 0.1804 \\ 0.2126 & 0.7151 & 0.0721 \\ 0.0193 & 0.1191 & 0.9503 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

The inverse transformation (CIE XYZ to RGB) is derived as follows:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2404542 & -1.5371 & -0.4985 \\ -0.9692 & 1.8760 & 0.0415 \\ 0.0556 & -0.2040 & 1.0572 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.2)$$

## 2.3 LMS

LMS color space depicts the response of the three types of cones in the human eye, sensitive to long (L), medium (M) and short (S) wavelengths. LMS is commonly used for performing a chromatic adaptation, *i.e.* adapting the appearance of colors under different illuminants. The chromatic adaptation is presented in following section. There exist several conversion matrices, transforming CIE XYZ into LMS, *e.g.* Bradford, CIECAM97 [10, 17], etc. In this dissertation, we use the CIECAM02 conversion matrix [10] as follows:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \mathbf{M}_{CAM02} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \text{ where } \mathbf{M}_{CAM02} = \begin{bmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{bmatrix} \quad (2.3)$$

When the CIECAM02 conversion matrix is used, the LMS color model is often referred to as spectrally sharpened RGB.

## 2.4 HSV

HSV is the cylindrical-coordinate representation of points in the RGB color space. Its channels are called hue, saturation and value (brightness). HSV represents a simple transformation of the RGB color model. HSV is more sensitive to changes in the brightness than to changes in the chroma. HSV is useful for extracting a single color from an image. However, this color space is not perceptual and it does not account for the complexity of the human perception.

## 2.5 CIE Lab

CIE Lab is a perceptual color space, designed to approximate the human color vision. The Euclidean distances between the colors in CIE Lab correspond to the perceptual color distances. The CIE Lab color model is device-independent, *i.e.* the colors remain constant

and independent of the device they are displayed on. CIE Lab color space has three dimensions, *i.e.* lightness and two color-opponents channels, based on the CIE XYZ color model. The first of the two color-opponent axes, *i.e.* the  $a$  axis, extends from green to red, while the second one, *i.e.* the  $b$  axis, extends from blue to yellow. The channels of the color space are independent, which makes CIE Lab suitable for a wide range of image color manipulations. CIE Lab is derived from CIE XYZ as follows:

$$L = 116f(X/X_w) - 16 \quad (2.4)$$

$$a = 500(f(X/X_w) - f(Y/Y_w)) \quad (2.5)$$

$$b = 200(f(Y/Y_w) - f(Z/Z_w)), \text{ where} \quad (2.6)$$

$$f(x) = \begin{cases} \sqrt[3]{x} & \text{if } x > \epsilon \\ \frac{\kappa x + 16}{116} & \text{otherwise} \end{cases} \quad (2.7)$$

with  $\epsilon = 0.008856$  and  $\kappa = 903.3$ . The triplet  $(X_w, Y_w, Z_w)$  corresponds to the CIE XYZ coordinates of the reference white point. The colors, produced by CIE Lab, are relative to the reference white point.

## 2.6 Lch

Lch color space is the polar coordinate representation of CIE Lab. The  $L$  channel, representing the lightness, is the same as the  $L$  channel of Lab. The circular axis  $h$  is the hue, whereas the  $c$  axis is the chroma, or the saturation. Lch is beneficial for a successful separation of the image colors one from another. We exploit this property of the Lch color model in chapter 5, part II. The channels of Lch are derived from CIE Lab as follows:

$$L = L \quad (2.8)$$

$$c = \sqrt{a^2 + b^2} \quad (2.9)$$

$$H = \begin{cases} \arctan(b/a) & \text{if } \arctan(b/a) \geq 0 \\ \arctan(b/a) + 360^\circ & \text{otherwise} \end{cases}, \quad (2.10)$$

where  $\arctan(x)$  denotes the arctangent function, calculated at  $x$ .

# Chromatic adaptation transform and white point

# 3

This chapter introduces the notion of white point and the method for adapting the colors of an image to a given white point, known as chromatic adaptation transform (CAT).

---

## 1 White point

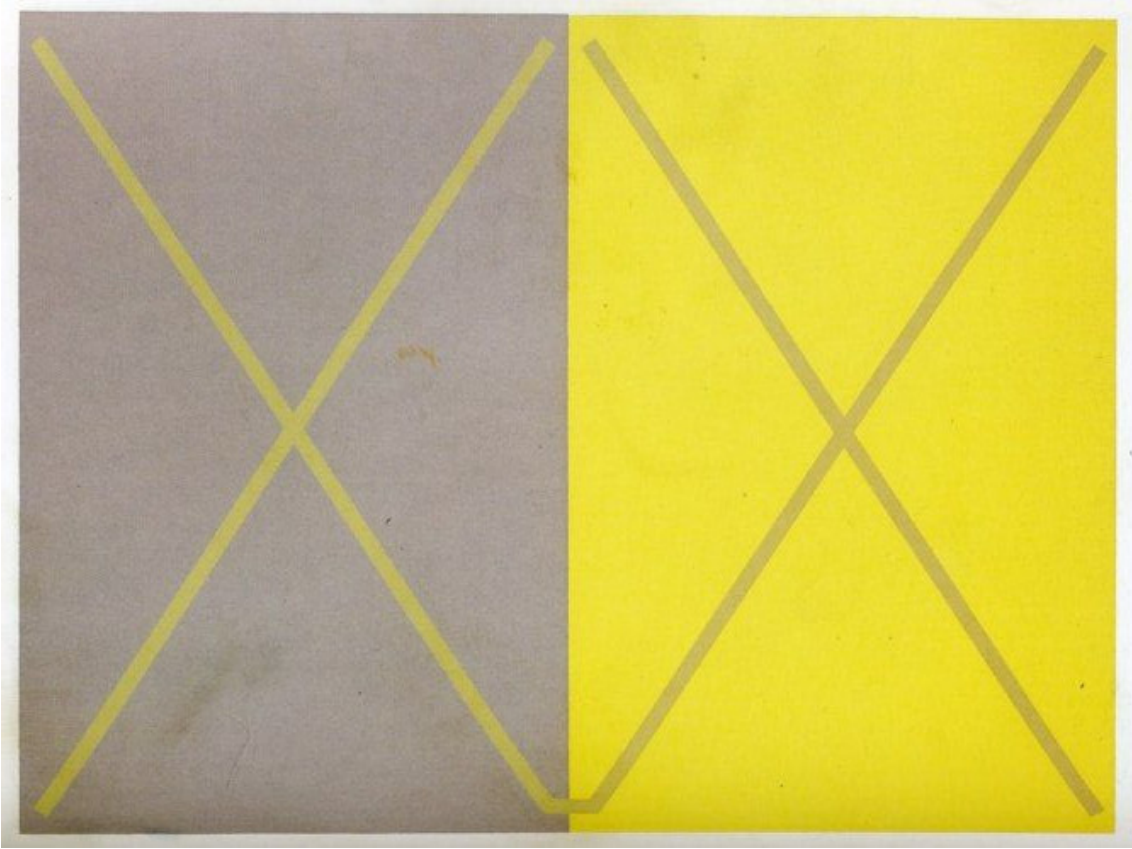
CAT adapts the colors of an input image to a given target illuminant. The target illuminant represents the white point of the target scene. The white point is a set of tristimulus values which defines the color of the light source in the scene. There exist a number of well-studied white points. For example, CIE D65 is the white point corresponding to the average daylight, whereas CIE D50 is the white point at sunrise or sunset [15, 16].

The white point of an image can also be computed using the color pixels of an image. Often, the white point is computed by assuming Gray World, *i.e.* the average of all the colors in an image is assumed to be neutral gray. Using this assumption, the color cast, caused by the illuminant in the scene, can be computed by comparing the average image color with neutral gray. Instead of averaging all the color pixels in an image, a more advanced solution is proposed in [18]. The method in [18] computes the white point as an average of all image pixels, similar to neutral gray. Removing the color cast, caused by the scene illuminant, is known as white balance [19]. White balance is one of the properties of the human eye. The human eye preserves the color constancy, *i.e.* the colors in a scene appear constant under different lighting conditions.

---

## 2 Chromatic adaptation transform

The color cast, introduced by an illuminant in a digital image, can be removed by performing the CAT. The CAT has been introduced as the first step of the color appearance model CIECAM02 [10]. Since then, the CAT has often been applied independently from CIECAM02 [7, 20]. In general, the CAT is employed to adapt the colors of an input image to a target illuminant. To this end, both the input image and the target illuminant are first converted to the LMS color space. The CAT consists of the following transformation,



**Figure 3.1** – The surrounding color plays a major role when perceiving a given color. The colors of the two crosses, though exactly the same, are perceived as different due to the different surrounding colors.

called von Kries normalization, carried out in LMS:

$$L_c = \left( L_{W_t} \frac{D}{L_{W_i}} + (1 - D) \right) L \quad (3.1)$$

$$M_c = \left( M_{W_t} \frac{D}{M_{W_i}} + (1 - D) \right) M \quad (3.2)$$

$$S_c = \left( S_{W_t} \frac{D}{S_{W_i}} + (1 - D) \right) S, \text{ where} \quad (3.3)$$

$$D = K \times S \times \left[ 1 - \left( \frac{1}{3.6} \right) e^{\left( -\frac{(L_A - 42)}{92} \right)} \right] \quad (3.4)$$

The triplet  $(L, M, S)$  denote each pixel of the input image in the LMS color space, whereas the triplets  $(L_{W_i}, M_{W_i}, S_{W_i})$  and  $(L_{W_t}, M_{W_t}, S_{W_t})$  are respectively the input and target white points. The input white point is always computed from the input image (for instance, by assuming Gray World), whereas the target illuminant is either given (*i.e.* it is a well-known illuminant) or is computed from a target image. The triplet  $(L_c, M_c, S_c)$  denotes the adapted color values in the LMS color space.

The factor  $D$  in equation 3.4 controls the level of adaptation to the target illuminant. The factor  $D$  ranges from 0 (no adaptation) to 1 (full adaptation). The scalar  $L_A$  is the

adapting luminance, taken as 20% of the white object in the scene. The surrounding factor, denoted by  $S$ , equals to 1 for average surround, 0.9 for dim surround and 0.8 for dark environments. The effect of the surround on the color appearance is illustrated in figure 3.1. Furthermore, a coefficient  $K = 0.3$  is used by Kuang et al. [21] to avoid full adaptation and de-saturation of the colors. In general, the coefficient  $K$  can vary from 0 to 1.





# Distribution models

# 4

The image features, such as the image color channels, the image gradient, etc. are modeled by well-known distribution models. Hereafter, we present some of the most common distribution models used in image processing.

## 1 Multivariate generalized Gaussian distribution

The MGGD belongs to the family of elliptical distributions. Often referred to as multivariate power exponential distribution, the probability density function  $f(\mathbf{x})$  of the MGGD is defined as follows [22]:

$$f(\mathbf{x}) = \frac{\Gamma(\frac{p}{2})}{\pi^{\frac{p}{2}}} h_{\Sigma, \beta}(\mathbf{x} - \boldsymbol{\mu}), \text{ where} \quad (4.1)$$

$$h_{\Sigma, \beta}(\mathbf{y}) = \frac{\beta}{2^{\frac{p}{2\beta}} |\Sigma|^{\frac{1}{2}} \Gamma(\frac{p}{2\beta})} \exp\left(-\frac{1}{2} (\mathbf{y}^T \Sigma^{-1} \mathbf{y})^\beta\right) \quad (4.2)$$

for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ ,  $p \in \mathbb{N}$ . The matrix  $\Sigma_{p \times p}$  is a positive definite scatter matrix, whereas  $\boldsymbol{\mu}$  and  $\beta$  are respectively the mean and the shape parameters of the distribution. Moreover,  $|\Sigma|$  denotes the determinant of matrix  $\Sigma$  and  $\Gamma(\cdot)$  is the gamma function [23]. The shape parameter  $\beta$  is connected to the sparseness of the distribution. The smaller the shape parameter, the sparser the distribution, *i.e.* the heavier its tails. Pascal *et al.* [24] have introduced the following parametrization of the scatter matrix:  $\Sigma = m\mathbf{M}$ , where  $m$  denotes the scale parameter of the MGGD and the matrix  $\mathbf{M}_{p \times p}$  is a positive definite scatter matrix, for which  $Tr(\mathbf{M}) = p$ ,  $Tr(\mathbf{M})$  denoting the trace of the matrix  $\mathbf{M}$ .

The MGGD has been used for modelling various features extracted from images, such as gradient fields [25] and wavelet coefficients [26]. The potential of the MGGD has been exploited for image processing tasks like image denoising [27, 28], texture classification [29], etc.

So far, several distance measures between two MGGDs have been tackled. The Kullback-Leibler divergence has commonly been used as a distance between MGGDs. To this end, Do *et al.* [30] propose a closed form of the Kullback-Leibler divergence for MGGDs, whereas Naornita *et al.* [31] generalize it for complex MGGDs. Furthermore, a more natural and more accurate similarity distance between MGGD probabilistic manifolds, namely the Rao-Kramer geodesic measure, has been introduced by Verdoolaege *et al.* [32]. A closed-form expression of the Rao-Kramer measure has been derived for a fixed shape parameter of the MGGD [32, 33]. The geodesic distance is used for texture classification and discrimination [32] whereas the Kullback-Leibler divergence is applied in the context of wavelet-based texture retrieval [30].

The following two sections are dedicated to two special cases of the MGGD, *i.e.* the multivariate Gaussian distribution and the multivariate Laplace distribution.

## 2 Multivariate Gaussian distribution

The multivariate Gaussian distribution (MGD) is a special case of the MGGD with a shape parameter equal to 1. The probability density function  $f(\mathbf{x})$  of a random vector  $\mathbf{x} \in \mathbb{R}^p$ ,  $p \in \mathbb{N}$ , distributed according to an MGD with a mean  $\boldsymbol{\mu}$  and a covariance matrix  $\boldsymbol{\Sigma}$ , is defined as follows:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right). \quad (4.3)$$

The covariance matrix  $\boldsymbol{\Sigma}$  defines the inter-component dependencies, whereas  $|\boldsymbol{\Sigma}|$  denotes the determinant of the matrix  $\boldsymbol{\Sigma}$ . The special case of a diagonal covariance matrix indicates an inter-component independence.

Furthermore, given a set of identically distributed random vectors  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ ,  $\mathbf{x}_i \in \mathbb{R}^p$ , following the law of an MGD with unknown parameters  $\boldsymbol{\Sigma}$  and  $\boldsymbol{\mu}$ , the maximum likelihood estimators (MLE) of the parameters  $\boldsymbol{\Sigma}$  and  $\boldsymbol{\mu}$  are computed as follows:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad (4.4)$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T (\mathbf{x}_i - \hat{\boldsymbol{\mu}}). \quad (4.5)$$

The MLE of  $\boldsymbol{\Sigma}$  is a biased estimator. Therefore, the coefficient  $1/N$  (equation (4.5)) is often replaced by  $1/(N-1)$  to obtain an unbiased estimator of  $\boldsymbol{\Sigma}$ .

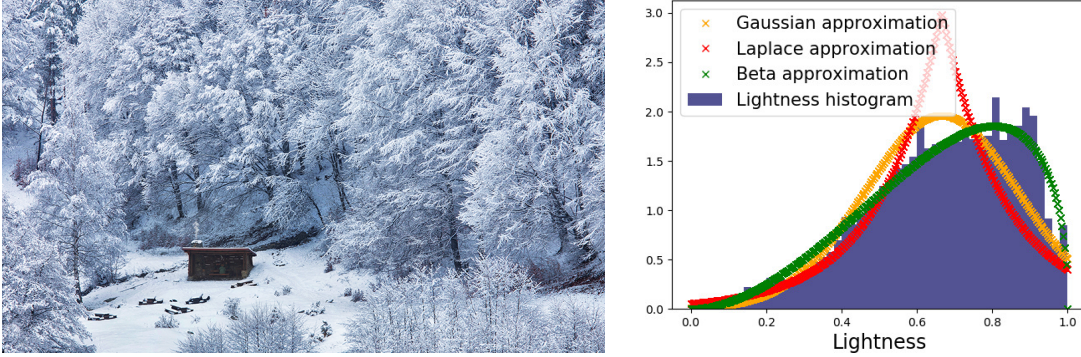
The MGD is one of the most popular distributions in image processing. Thanks to its relative simplicity and analytical properties, the MGD is often chosen to model image color distributions [34]. The MGD is characterized by lighter tails (lighter than an MGGD with a shape parameter less than 1) and thus, it does not describe well enough heavy-tailed distributions, such as image gradient distributions.

A single MGD may not fit any image distribution. When a distribution cannot be fitted well enough by a single MGD, it is often fitted by a mixture of MGDs [7]. Gaussian mixture models (GMM) are used to describe such complex distributions. GMM represent the unknown distribution as a mixture of several Gaussian distributions. To compute the parameters of each mixture, the expectation-maximization (EM) algorithm is used. GMM are often adopted to perform a soft segmentation of the image color channels [5, 7].

## 3 Multivariate Laplace distribution

The multivariate Laplace distribution (MLD), also known as double exponential distribution, is a continuous distribution which has heavier tails than those of the MGD. It is a special case of the MGGD with a shape parameter equal to 0.5. The probability density function  $f(\mathbf{x})$  of a random vector  $\mathbf{x} \in \mathbb{R}^p$ ,  $p \in \mathbb{N}$ , distributed according to an MLD with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ , is defined as follows:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{(p/2)}} \frac{2}{\lambda} \frac{K_{(p/2)-1} \left( \sqrt{\frac{2}{\lambda} q(\mathbf{x})} \right)}{\left( \sqrt{\frac{\lambda}{2} q(\mathbf{x})} \right)^{(p/2)-1}}, \text{ where} \quad (4.6)$$



**Figure 4.1** – The lightness histogram of the left-hand image is right-skewed (as shown in the right-hand plot). Being both continuous symmetrical distributions, the MGD and the MLD fail to describe the skewness of the lightness histogram. In contrast, the bounded Beta distribution provides the best model for the image lightness distribution by accounting for its right skewness.

$$q(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}). \quad (4.7)$$

The parameter  $\lambda$  denotes the scale parameter of the MLD, whereas  $K_l(\mathbf{x})$  denotes the modified Bessel function [35] of second kind and order  $l$ , evaluated at  $\mathbf{x}$ . The MLD is usually adopted to fit sparse distributions. The MLD is used to model image gradient distributions [36] and has proved beneficial for speech recognition [37].

## 4 Beta distribution

Both the MGD and the MLD, and in general the MGGDs, are continuous symmetrical distributions. To this end, they are not suitable for modelling skewed distributions. Figure 4.1 shows that, often, the image color channels are not symmetrical but rather skewed. To account for the skewness of the image feature distributions, the Beta distribution can be employed.

The density function  $f(\cdot)$  of a Beta distributed random variable  $\mathbf{x}$  with shape parameters  $\alpha, \beta > 0$  (denoted  $\mathbf{x} \sim \text{Beta}(\alpha, \beta)$ ) is given as follows:

$$f(\mathbf{x}) = \frac{1}{\mathbf{B}(\alpha, \beta)} \mathbf{x}^{\alpha-1} (1 - \mathbf{x})^{\beta-1}, \quad (4.8)$$

where  $\mathbf{x} \in [0, 1]$ , and  $\mathbf{B}(\alpha, \beta)$  denotes the Beta function.

Let  $\mu$  and  $\sigma^2$  be respectively the sample mean and variance of a Beta distribution. Then, approximations of the shape parameters  $\alpha$  and  $\beta$  are given as functions of  $\mu$  and  $\sigma^2$ :

$$\alpha = \mu^2 \left( \frac{1 - \mu}{\sigma^2} - \frac{1}{\mu} \right), \quad (4.9)$$

$$\beta = \alpha \left( \frac{1}{\mu} - 1 \right). \quad (4.10)$$

More precise estimation (from samples) of the shape parameters  $\alpha$  and  $\beta$  can be obtained by applying the expectation-maximization (EM) algorithm.

The Beta distribution can admit different shapes and thus, describe a variety of unknown distributions. In particular, the Beta distribution describes very well the skewed distributions, as shown in figure 4.1. Furthermore, in comparison with the MGD and the MLD, the Beta distribution is a bounded distribution and therefore, describes well discrete distributions such as the distributions of color and light in images. Furthermore, Beta mixture models (BMM) have proved more efficient than GMM in areas like recognition of handwritten digits [4] and skin segmentation [38].

## **Part II**

### **Distribution-based transfer for example-guided image editing**



# Introduction

In this part, our three new color transfer methods are detailed. Color transfer aims to modify the look of an input image according to the illumination and the color palette of a target image. Color transfer is often viewed as a distribution transfer problem, in which the multivariate Gaussian distribution (MGD) plays a significant role. Early research in the field of color transfer assumes that the color and light distributions of images follow a Gaussian distribution. This assumption has been beneficial for computing a number of global Gaussian-based transformations [3, 7, 34]. When the Gaussian assumption does not hold, such global transformations fail to properly transfer the target colors and may produce results inconsistent with the target image specifics.

According to Reinhard et al. [3] and more recently, to Bonneel et al. [6], the limitations of global transformations could be overcome with cluster-based techniques. In chapter 5, we propose a local method which uses Gaussian mixture models (GGM) to fit the image distributions by partitioning the images into several clusters. Our key idea consists in determining which feature, *i.e.* color or light, is more representative for a given image. Then, we use this feature to cluster the image. We design four novel policies for mapping the input and target image clusters. These policies contribute to producing photo-realistic images, the style of which is similar to the style of the target image (as a remark, in our context, the style of images is defined in terms of color and light). Our style-aware color transfer method is presented in chapter 5.

Apart from color and light, gradient is another image feature that impacts the perception of images. The distribution of an image gradient is a heavy-tailed distribution and therefore, it cannot be fitted by the MGD. To be able to describe the distributions of both color and gradient by a single model, we adopt the multivariate generalized Gaussian distribution (MGGD). Including the MGD and the MLD as special cases, the MGGD with an unconstrained (by a fixed value) shape parameter is likely to accurately fit a wide class of image feature distributions, including those of color, gradient, wavelet coefficients, etc. Chapter 6 introduces our transformation between two MGGDs. Our MGGD transformation is applied in the context of color and gradient transfers and it also carries out a multidimensional simultaneous transfer of color and gradient.

The MGGD and in particular its special cases, *e.g.* the MGD and the MLD, belong to the class of unbounded distributions. Despite that, they have been commonly used to describe bounded distributions, such as the distributions of color and light. We address this issue in chapter 7. We adopt the Beta distribution to model the color and light distributions of images and propose a Beta-based transformation, which is applied in the context of color transfer. The results in chapter 7 illustrate the advantage of modeling color and light by bounded distributions to using unbounded (Gaussian) distributions to fit color and light.

Different color transfer methods often result in different output images. The process



of determining the most plausible output image may be subjective, as it depends on a person's preference. To lessen the level of subjectivity in quality assessment for color transfer, we propose a model for objective evaluation of the color transfer quality. Our model explains the relationship between users' perception and a number of image features. The model is introduced in [chapter 8](#).

# Style-aware robust color transfer

# 5

---

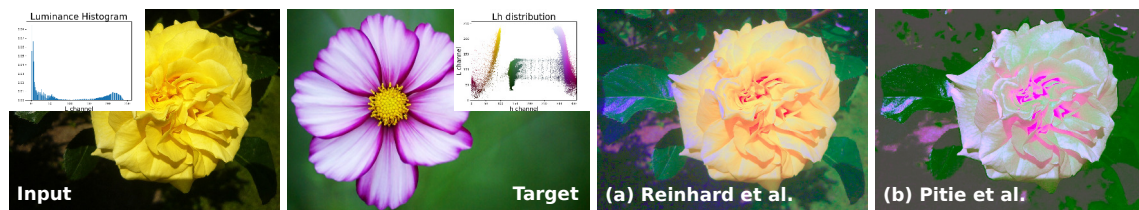
## 1 Introduction

In this chapter, we propose an original way of carrying out a color transfer between input and target images by respecting the style of the target image. We call this process a style transfer. We define style of images as a combination of two key image features: light and colors. The style transfer strongly depends on these two image features. Global methods transfer the key characteristics (*e.g.* mean, standard deviation) of the target color and light distributions. However, a global distribution transfer may be insufficient to grasp local variations of light and color (for an illustration, see figure 5.1). This is due to the fact that the efficiency of most global color transformations relies on the validity of an assumption. Global color transfer methods assume that the color and light distributions of the input and target images can be fitted well by a single MGD. However, in general, the global Gaussian assumption turns out to be too restrictive to ensure a good color transfer. In many cases, results from a global color transfer are inconsistent with the style of the target image and at odds with the concept of a balance between the different features in an image. Such an imbalance may, for instance, cause over-/under-saturation or images of low visual quality, as illustrated in figure 5.1.

We can overcome the aforementioned limitations with cluster-based techniques. In this case, color transfer is performed between clusters in images rather than between entire images. In this chapter, we propose a local style-based method which uses GMM to describe the color and light image distributions. Our method partitions the images into several clusters according to the key feature of each image. We design a classification method, which automatically determines the most important feature in an image. Once the image clustering is performed, our method locally applies an optimal color transformation between corresponding clusters from the input and target images. Our results look more natural and aesthetically pleasing than results from state-of-the-art methods. Moreover, a user study has shown that our results match closely the user's expectation of a style transfer.

The main contributions, presented in this chapter, are threefold:

- a novel automatic system that classifies an image as a light-based image or a colors-based image and determines the input/target clusters;
- a cluster-based method consisting of several novel mapping policies between the input/target clusters;
- a subjective user evaluation of the results as well as an objective evaluation consisting of two objective metrics.



**Figure 5.1** – Two results, obtained with two global color transfer methods. The global transformations assume that the light and color distributions of the input and target images follow the Gaussian law. This assumption does not hold for the input and target images, shown in this figure. The input image consists of two luminance clusters (corresponding to the two peaks in its luminance histogram), whereas the target image is composed of three color clusters (as shown in its luminance-hue plot). Therefore, the distributions of the input luminance and the target colors cannot be fitted by a single MGD. Due to this fact, the global color transformations fail to produce plausible results. Result (a) does not match the target colors, whereas result (b) is significantly over-saturated, which compromises its photo-realism. Moreover, result (b) does not respect the target style specifics in terms of light and color (for instance the foreground color in result (b) does not match precisely the target foreground color).

The chapter is organized as follows. Section 2 presents the related research work. Section 3 provides details on our classification system as well as on the color grading method, used in this paper. Moreover, section 3 also introduces our four novel mapping policies. Results, user evaluation and objective evaluation are presented in section 4. Section 5 shows limitations and provides ideas for future research work. Finally, the last section concludes the chapter.

**Publication.** The work, presented in this chapter, has been published in the following paper:

*H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, “Style-aware robust color transfer”, in Proceedings of the workshop on Computational Aesthetics. Eurographics Association, 2015, pp. 67-77.*

## 2 Related work

This section is divided into two parts. First, we present state-of-the-art color transfer methods. Second, as the goal of this chapter is to introduce our method for style transfer, we present recent research works in connection with the artistic style transfer between images (mainly based on convolutional neural networks).

### 2.1 Color transfer

Research works in the field of color transfer are classified by Faridul [1] into two main categories: geometry-based and statistical-based methods. Geometry-based approaches aim at finding content-based correspondences in pairs of images and ensure that these correspondences have the same colors.

We prefer statistical-based methods to geometry-based ones because the latter heavily depend on the image contents and therefore, it is not an easy task to apply them to image

pairs with various contents. The method, presented in this chapter, focuses on statistical-based techniques in order to carry out a color transfer between images of various contents and styles. Generally, we could divide the statistical-based methods into two classes: non-parametric and parametric (distribution-based) methods. The non-parametric methods carry out the transfer of color and light regardless of the type of input and target distribution. In contrast, the parametric methods apply color transformations under the assumption that the color distributions of the input and target images follow a particular distribution.

### 2.1.1 Non-parametric methods

The naive histogram matching is the first example of a non-parametric method. It attempts to borrow the thorough look of an image by matching two cumulative density functions. However, a full histogram transfer tends to be too harsh and may result in artifacts. Therefore, recent works resolve this problem by matching histograms at different scales [39,40] and by performing gradient optimizations [41]. Additionally, Pitié et al. [42] match two 3D distributions through rotations and 1D histogram projections. Their method manages to entirely transform the input distribution to the target distribution.

Recently, three papers have introduced new non-parametric approaches for coloring an image [20,43,44]. Hwang et al. [44] employ a non-linear interpolation using probabilistic moving least squares. The authors apply a color transfer to images of the same scene, taken under different lighting conditions. In contrast, Frigo et al. [20] apply their method to images with various contents. Frigo's non-parametric approach is based on an optimal transportation problem that minimizes the overall displacement cost of the color mapping. Moreover, Frigo et al.'s method makes no assumptions about the color distributions of the input and target images (unlike, for instance, Pitié et al.'s method [34], which assumes that the color distributions follow the Gaussian law). Furthermore, due to smooth interpolation, Frigo's method could create colors which do not appear in the target image. In contrast, Nguyen et al. [43] argue that a good color transfer is a color transfer which does not result in out-of-gamut colors. Their illuminant-aware color transfer produces images with colors inside the target gamut.

### 2.1.2 Parametric methods

The first parametric color transformation has been introduced by Reinhard et al. [3]. This method fits the distribution of color by an MGD with a diagonal covariance matrix. The transformation is then carried out independently on each component of the  $L\alpha\beta$  color space. Reinhard et al.'s method has been specially designed for natural images, as in the majority of cases, their color and light distributions are similar to the MGD.

Reinhard et al.'s method is suitable for image distributions with independent color components. To account for inter-component dependencies, color and light distributions are assumed to follow an MGD with non-diagonal covariance matrix. This assumption has been adopted in a number of color transformation, including the Cholesky decomposition [34], and the square root decomposition, also known as principal component analysis (PCA) [45]. Both transformations have been applied in the context of color transfer [34]. In particular, a PCA-based transfer has been applied to gray-scale image coloring [46] and fuzzy color transfer [45].

In applications, for which the geometry of the transformed distribution is of a great significance, as it is the case of color transfer, both the Cholesky decomposition and the square root decomposition encounter problems [34]. To preserve the geometry of the resulting distribution as intended, Pitié et al. [34] propose a transformation based on a minimization of the displacement cost for transforming an MGD into another one. The minimization problem is well-known as the Monge-Kantorovich transportation problem [47] with a solution, given by the linear closed-form Monge-Kantorovich transformation [34, 48]. The significance of the Monge-Kantorovich mapping for color transfer is presented in [34].

Recently, the Monge-Kantorovich optimal transformation has received much attention in connection with the local color transfer method by Bonneel et al. [6]. The method applies the Monge-Kantorovich transformation locally between corresponding input and target clusters. To obtain the image clusters and map the input clusters to the target clusters, Bonneel et al. use a luminance-based approach. In contrast, Tai et al. [5] apply a 3D EM algorithm to cluster the input and target images. Then, they build a mapping function based on the mean luminance of the input and target images.

### 2.1.3 Limitations

Hereafter, we discuss some key limitations of both global and local color transfer methods.

Global color transfer methods rely on the assumption that the MGD can fit the entire color and light distributions the input and target images. This assumption turns out to restrict the color transfer, as it will be illustrated in section 4.

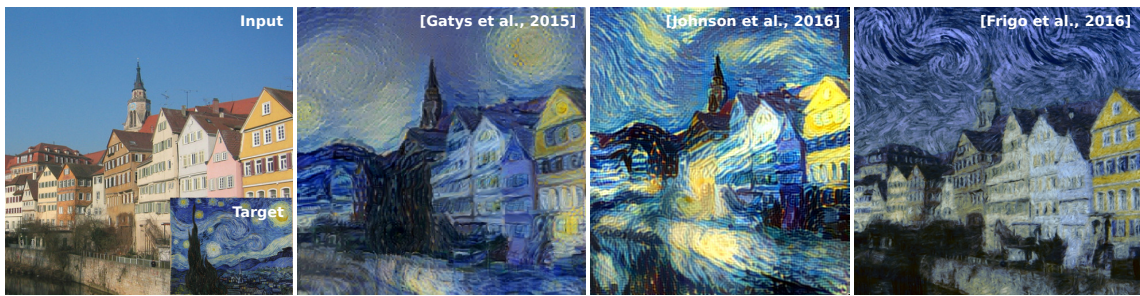
In contrast, local color transfer methods target only images with similar low-level characteristics. For instance, Bonneel et al.’s method [6] would always segment the input and target images into three luminance clusters regardless of whether the images contain enough luminance information for such a segmentation or not. Furthermore, the local color transfer methods do not elaborate enough on choice of the mapping functions linking the input and target clusters. For example, there is no justification if the chosen luminance-based mapping policies are optimal and what their advantages and limitations are. Finally, a good method should jointly consider light and color features (as opposed to only light for the clustering).

Unlike the existing local methods in the field of color transfer, we take into account both characteristics in the clustering process and introduce novel style-based mapping policies which, to the best of our knowledge, have not been considered so far.

---

## 2.2 Style transfer

Recently, a breakthrough in the context of transferring the artistic style of a target image to an input image has been made. Several style transfer methods based on deep convolutional neural networks, have been proposed. Convolutional neural networks [49] efficiently detect image features at small and large scales. Neural networks are trained to extract the most important image features content-wise and to produce feature maps at different convolution layers. The feature maps in the convolutional layers represent the content of a given image at various scales. As we go deeper in the network, the convolutional layers represent higher-level content. By using non-linear combinations of the detected features,



**Figure 5.2** – Artistic style transfer (carried out with three different methods) using a painting as a target image. The methods transfer the target style using convolutional neural networks (first two results, left to right) and specially adaptive image partitioning (the right-most result).

the convolutional neural networks can recognize objects in images and videos and thus, classify them [50, 51]. Hereafter, we discuss methods which adopt convolutional neural networks to represent the content and the style of a given image and to carry out an artistic style transfer between two images.

### 2.2.1 Non-photo-realistic style transfer

Gatys et al. [52] adopt the convolutional neural networks to represent the content of an input image and the style of a target image and perform a style transfer between the two images. They extract a representation of the style by computing the correlation of the values within a given feature map. Gatys et al.’s style transfer is presented as an optimization problem aiming to minimize a loss function. The loss function of the method is a combination of a style loss, *i.e.* the distance between the style representations of the result and the target image, and a content loss, *i.e.* the distance between the content representations of the result and the input image. The stylized result is generated through a gradient descent via backpropagation [53]. Gatys et al.’s method is used to transfer an artistic target style (usually a painting) to an input image (usually a photograph) as shown in figure 5.2.

One of the main drawbacks of Gatys et al.’s method is the time it takes to generate a stylized image (the stylization process may take up to several hours). To this end, Johnson et al. [54] optimize the original algorithm by Gatys et al. so that it runs in real-time. Johnson et al. add a step, called image transformation network, to transform the input image through series of convolutional down- and up-sampling. The transformed input image is then given to the loss function optimization in [52]. Johnson et al.’s framework reduces significantly the computational time and produces qualitatively similar results to those from Gatys et al.’s method (figure 5.2).

Apart from the neural networks, Frigo et al. [55] propose a style transfer method based on a local texture transfer using a spatially adaptive partition of the input image. The authors introduce a “Split and Match” image decomposition which is guided by the similarity between the input and target patches. Figure 5.2 compares Frigo et al.’s method to the style transfer methods using convolutional neural networks.

The three formally discussed methods perform a non-photo-realistic style transfer. They produce impressive results for target images which are paintings and sketches. However, they are not suitable for a photo-realistic style transfer, as they introduce many image



Notation	Definition
$CS \in \{rgb, lab, lch\}$	RGB, Lab or Lch
$\mathbf{I}_{CS} : \Omega_{\mathbf{I}} \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$	input image
$\mathbf{J}_{CS} : \Omega_{\mathbf{J}} \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$	target image
$\mathbf{O}_{CS} : \Omega_{\mathbf{I}} \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$	output image
$\mathbf{R} \in \{\mathbf{I}, \mathbf{J}\}$	image $\mathbf{I}$ or image $\mathbf{J}$
$\Omega_{\mathbf{R}}$	the set of spatial coordinates of the image $\mathbf{R}$
$m_{\mathbf{R}}$	number of clusters of $\mathbf{R}_{CS}$
$N$	$\min(m_{\mathbf{I}}, m_{\mathbf{J}})$
$R_{CS}^{(k)}$	$k^{th}$ cluster of $\mathbf{R}_{CS}$
$\zeta_{CS}^{\mathbf{R}} = \{\mathbf{R}_{CS}^{(1)}, \dots, \mathbf{R}_{CS}^{(m_{\mathbf{R}})}\}$	set of clusters of $\mathbf{R}_{CS}$
$\zeta_{CS}^{\mathbf{R}}(v)$	$v^{th}$ component of $CS$ in $\zeta_{CS}^{\mathbf{R}}$
$f_{CS}^{(k)}$	pdf of $\mathbf{I}_{CS}^{(k)}$
$g_{CS}^{(k)}$	pdf of $\mathbf{J}_{CS}^{(k)}$
$p_{ik}$	probability of the $i^{th}$ pixel to belong to $\mathbf{R}_{CS}^{(k)}$
$\mathbf{O}_{CS}^{(k)}$	$k^{th}$ cluster of $\mathbf{O}_{CS}$

Table 5.1 – Correlations between the style and aesthetics scores.

distortions [56].

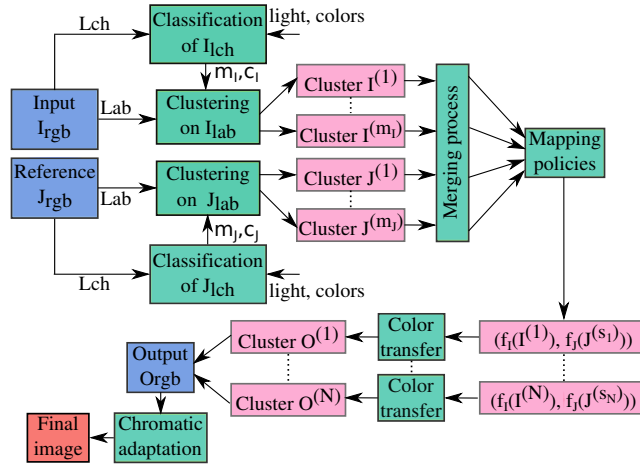
### 2.2.2 Photo-realistic style transfer

To adapt the convolutional neural networks for a photo-realistic style transfer, Luan et al. [56] propose a regularization which constrains the transformation from the input image to the output image. Luan et al. add a regularization term to the optimization to penalize image distortions and ensure an affine transformation between the input and output image patches. To further improve the photo-realism of the style transfer, the authors adopt a semantic segmentation of the input and target images. Luan et al. show that their method outperforms the methods in [52, 57] (based on convolutional neural networks) for a photo-realistic style transfer. Luan et al.’s style transfer produces less satisfying results when the semantics of the input and target images are too distant, as the great mismatch comprises the photo-realism of deep learning style transfers [56].

The following section presents our style-aware color transfer method, which considers the local variations of light and color in images without accounting for the image content. This makes our method suitable for images with various contents and semantics. At the same time, our method proves to be more robust than state-of-the-art color transfer methods when it comes to representing target style features in terms of light and colors. We apply our method in the context of photo-realistic style transfer.

## 3 Style-aware style transfer

Our method is designed to carry out a local image color transfer by taking into account the main features of the input and target images. Our method focuses on light and colors



**Figure 5.3** – The framework illustrates the steps of the proposed method (the green boxes). The blue boxes stand for images in the RGB color space and in violet are the image clusters.

as the two key features of the image style. These features are involved in the process of clustering the input image  $I$  and the target image  $J$ . As a result of the clustering, we obtain a set of clusters for both images. We propose four policies to map the input and target clusters. The appropriate (for a given image pair) mapping policy and the number of input and target clusters are automatically determined and depend on the main features (light or color) of the input and target images. We apply an optimal transformation to the colors of each pair of corresponding clusters. In our method, the luminance and the colors are treated differently. To obtain the final result, we apply a local chromatic adaptation transform on the colorized (using the optimal transformation) image, which affects the luminance of the image (and the colors, as a consequence). Our method is illustrated in figure 5.3 and the main notations used in this chapter are given in table 5.1. Hereafter, we present the steps of our method in more details.

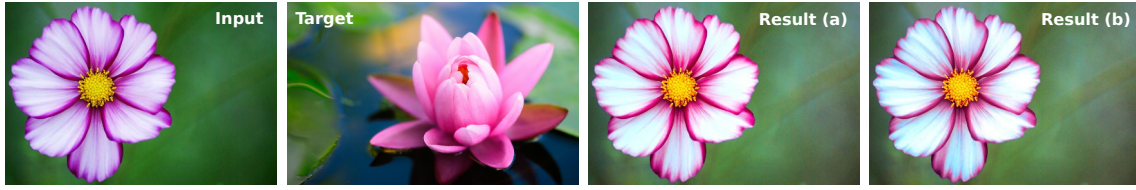
### 3.1 Automatic light-based versus colors-based classification

The proposed automatic classification system determines which feature, among the two considered features, *i.e.* light and colors, carries more information about the style of an image. The modification of these two features influences the thorough look and perception of images [44]. Therefore, if we modify the light and the colors of an image in a specific way, we would get close to a given image style.

We propose to classify the images into two main categories: colors-based style images and light-based style images (refer to figure 5.4 for illustration). The two types of images are defined hereafter.

- Colors-based style images are images whose color information is sufficient enough to define well at least two different and significant colors. An image of only one color is not considered as a colors-based style image. One color is not representative of the image style and there is a high probability that the light feature of that image has a greater impact on its style.
- Images which are not classified as colors-based style images are classified as light-based style images because their light features are more meaningful than their color





**Figure 5.4** – Result (a) is obtained with the default values of the three parameters of our classification system, whereas result (b) is obtained by setting  $d_p$  to half of its default value.

features.

The classification algorithm is performed in the Lch color space on both the input and target images. It consists of three steps:

1. Image  $\mathbf{I}_{rgb}$ <sup>1</sup> is first converted into the Lch color space to obtain image  $\mathbf{I}_{lch}$ . The set  $\mathbf{G}$  of gray points is extracted from image  $\mathbf{I}_{lch}$  as follows:

$$\mathbf{G} : \{(l, c, h) \mid c < c_{min}, \forall (l, c, h) \in \mathbf{I}_{lch}\} \quad (5.1)$$

where  $c_{min}$  is the threshold for the chroma component of the Lch color space.

2. The hue histogram function  $\phi : \Omega_\phi \subset \mathbb{R} \mapsto \mathbb{R}$ , where  $\Omega_\phi$  is the set of bin values in the Lch color space, is computed from the set of the remaining non-gray pixels denoted as  $\mathbf{I}_{lch} \setminus \mathbf{G}$ . We perform a linear search to obtain the local peaks of the hue histogram. Let  $\mathbf{P}$  be the set of all the local peaks in the hue histogram and let  $\delta$  define a small neighbourhood around a peak  $p \in \mathbf{P}$ . Then, the set of significant peaks  $\mathbf{P}_s$  is defined in the following way:

$$\forall p \in \mathbf{P} : p \in \mathbf{P}_s \Leftrightarrow \begin{cases} \sum_{\epsilon=-\delta}^{\delta} \phi(p + \epsilon) > s_{min}, \text{ and} \\ |p - p_s| > d_p, \forall p_s \in \mathbf{P}_s \end{cases} \quad (5.2)$$

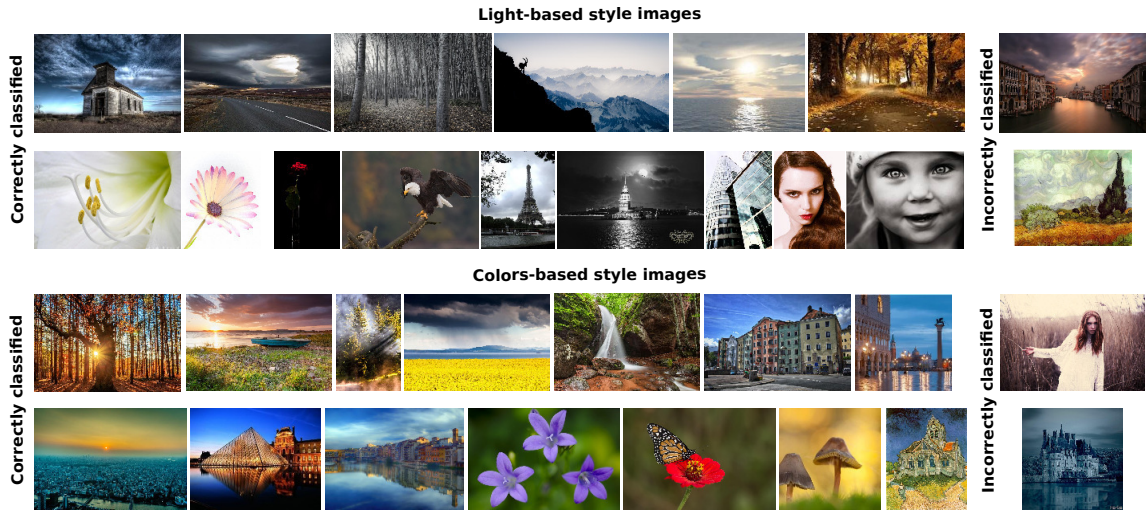
where  $s_{min}$  is the threshold for the minimum number of pixels defining a significant peak and  $d_p$  is the distance threshold between two significant peaks in the hue histogram.

3. Finally, if  $|\mathbf{P}_s| > 1$ , the original image is considered as a colors-based style image. Otherwise we classify the original image as a light-based style image.

We experimentally set the three parameters  $c_{min}$ ,  $s_{min}$  and  $d_p$  to default values (refer to section 3.5 for more details), although they can influence the final result as illustrated in figure 5.4.

The advantages of our classification algorithm lie in its simplicity and efficiency. We tried out the system on several image categories: natural scenes, city and street images, macro images, studio images and paintings. Figure 5.5 show images, which are correctly and incorrectly classified by our classification system. The judgment depends on the definitions of colors-based and light-based style images and it is partially based on intuition.

<sup>1</sup>In the description of the classification algorithm,  $\mathbf{I}_{rgb}$  refers to either the input or target images.



**Figure 5.5** – Examples of images correctly and incorrectly classified as light-based style images (first two rows) and colors-based style images (second two rows) respectively.

## 3.2 Clustering

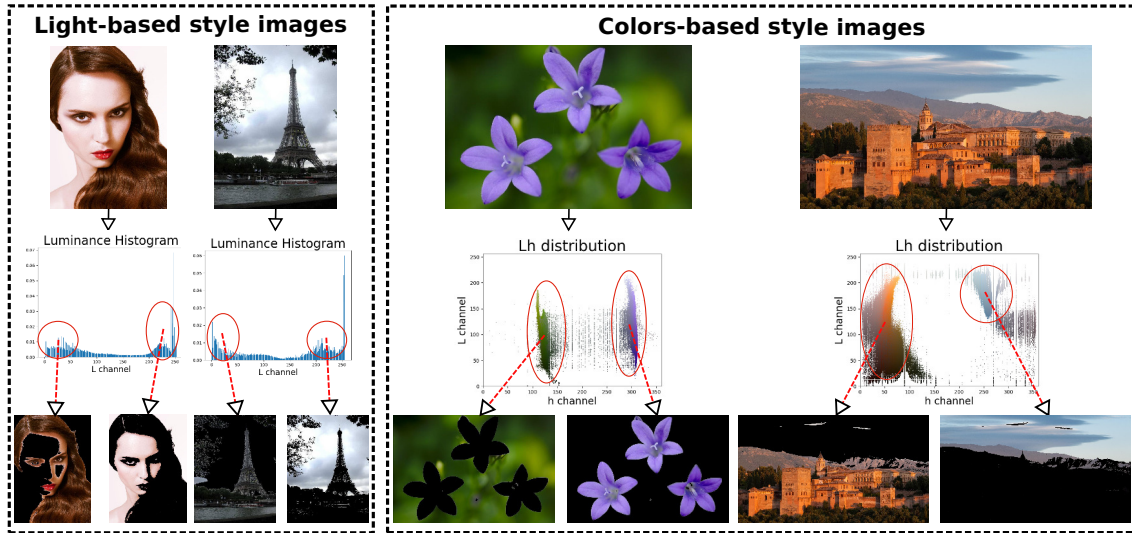
Once the main features of the input and target images are determined, a clustering is performed on both images. The number of the clusters, into which an image is segmented, is determined automatically. If an image is classified as a colors-based style image, the number of clusters is computed as the number of the significant peaks in its hue histogram, *i.e.*  $|\mathbf{P}_s|$ . For a light-based style image, we adopt Bonneel’s idea [6] of using luminance clusters, namely highlights, midtones and shadows. As a remark, there exist images for which one of these luminance clusters is not significant. In that case, we consider only two clusters as shown in figure 5.6. The number of the significant luminance clusters is equal to the number of the significant peaks in the luminance histogram.

The clustering is performed using the EM algorithm in the Lch color space. Figure 5.6 illustrates the two types of image clustering. For light-based style images, we choose to perform the clustering on the luminance histogram as in the majority of the cases it is more meaningful than the hue histogram. For colors-based style images, we choose the 2D Luminance-Hue distribution as it represents well the different color clusters of colors-based style images, as shown in figure 5.6.

Note that the number of input and target clusters can be different. In that case, if the number of input clusters is higher than the number of target clusters, the most similar input clusters (in terms of their means) are merged together (the case when the number of target clusters is higher than the number of input clusters is analogical). At the end of the clustering process, we obtain  $N$  clusters for both input and target images.

## 3.3 Mapping policies

As a next step, a mapping function between the input and target clusters is built. Hereafter, we introduce four mapping policies (for four different types of image pairs), namely Light to Colors, Colors to Light, Light to Light and Colors to Colors. The pseudo code in algorithm 1 illustrates the general logic behind of our mapping policies. In the following



**Figure 5.6** – The luminance histograms of the light-based style images have two significant peaks, which define two luminance clusters. The 2D Luminance-Hue distributions of the colors-based style images clearly separate the two main colors in the images. Overlapping is included when visualizing the clusters (the overlapping is more evident for the light-based style images).

subsections, we discuss the mapping policies.

### 3.3.1 Light to Colors

Light to Colors policy is designed for images with distant styles. The input image is a light-based style image, whereas the target image is a colors-based style image. As it will be presented in section 4, such test cases are challenging for the state-of-the-art methods. To deal with this issue, we developed a meaningful mapping function which links the light features of the input image to the color features of the target image.

Usually people expect cold colors to be present in the shadows of an image, whereas warm colors are likely to appear as highlights. The majority of photographers use the same approach as an artistic effect [58]. They use cold colors to indicate shadows or background, and warm colors to highlight bright areas, as shown in figure 5.7. Our Light to Colors mapping function is based upon these two artistic approaches.

The policy starts by selecting which cluster among the set of input clusters  $\zeta_{lch}^I$  has the minimum average luminance value. This problem is handled by the function  $FindDarkestCluster(\zeta_{lch}^I(l))$ , the output of which is the index  $s_I$  of the input cluster with the minimum average luminance value, and the pdf  $f_{lab}^{(s_I)}$  (of the selected input cluster). Similarly, the function  $FindColdestCluster(\zeta_{lch}^J(h))$  returns the index  $s_J$  of the target cluster with the maximum average hue value, and the pdf  $g_{lab}^{(s_J)}$  (of the selected target cluster). Our algorithm adopts the standard hue wheel [59]. The warmest color among a set of colors is defined as the one with the lowest hue value, whereas the coldest color as the one with the highest hue value. For instance, red and orange are considered warm colors, whereas blue and purple are considered cold colors. The function  $FindColdestCluster(\zeta_{lch}^J(h))$  finds the target cluster associated with the coldest color in the set of clusters  $\zeta_{lch}^J$ .

Once computed, both probability functions  $f_{lab}^{(s_I)}$  and  $g_{lab}^{(s_J)}$ , defined in the Lab color

**Algorithm 1** Mapping policies

---

```

1: for  $k = 1, \dots, N$  do
2:   if Light to Colors then
3:      $[s_I, f_{lab}^{(s_I)}] = \text{FindDarkestCluster}(\zeta_{lch}^I(l))$   $\triangleright$  find the input cluster with the lowest luminance
4:      $[s_J, g_{lab}^{(s_J)}] = \text{FindColdestCluster}(\zeta_{lch}^J(h))$   $\triangleright$  find the target cluster with the highest hue
5:   if Colors to Light then
6:      $[s_I, f_{lab}^{(s_I)}] = \text{FindColdestCluster}(\zeta_{lch}^I(h))$   $\triangleright$  find the input cluster with the highest hue
7:      $[s_J, g_{lab}^{(s_J)}] = \text{FindDarkestCluster}(\zeta_{lch}^J(l))$   $\triangleright$  find the target cluster with the lowest luminance
8:   if Light to Light then
9:      $[s_I, f_{lab}^{(s_I)}] = \text{FindDarkestCluster}(\zeta_{lch}^I(l))$   $\triangleright$  find the input cluster with the lowest luminance
10:     $[s_J, g_{lab}^{(s_J)}] = \text{FindDarkestCluster}(\zeta_{lch}^J(l))$   $\triangleright$  find the target cluster with the lowest luminance
11:   if Colors to Colors then
12:      $[s_I, s_J, f_{lab}^{(s_I)}, g_{lab}^{(s_J)}] = \text{FindMinDistPair}(\zeta_{lch}^I(h), \zeta_{lch}^J(h))$   $\triangleright$  find the two most similar input and target clusters (in terms of hue)
13:      $[\%_{lab}^{(k)}] = \text{PerformTranspOnAB}(f_{lab}^{(s_I)}, g_{lab}^{(s_J)})$   $\triangleright$  perform the color transformation on the chroma channels
14:      $\zeta_{lch}^I \setminus \mathbf{I}_{lch}^{(s_I)}$ 
15:      $\zeta_{lch}^J \setminus \mathbf{J}_{lch}^{(s_J)}$ 
16:  $[\mathbf{O}_{rgb}^{final}] = \text{CATLocal}(\mathbf{O}_{rgb}, \mathbf{J}_{rgb})$   $\triangleright$  perform local CAT

```

---

space, are passed to the function *PerformTranspOnAB*( $\cdot$ ). This function carries out the color transformation between the clusters  $\mathbf{I}_{lab}^{(s_I)}$  and  $\mathbf{J}_{lab}^{(s_J)}$ , which will be presented in section 3.4. Finally, the clusters  $\mathbf{I}_{lab}^{(s_I)}$  and  $\mathbf{J}_{lab}^{(s_J)}$  are removed from the sets  $\zeta_{lch}^I$  and  $\zeta_{lch}^J$  respectively. We repeat the described procedure for each input cluster.

At the end of algorithm 1, our Light to Colors policy will have mapped the warmest colors of the target image to the highlights of the input image and reversely, the coldest colors of the target image to the shadows of the input image. Given this mapping function, we are able to carry out a transfer between a light-based style image and a colors-based style image. To the best of our knowledge, there exists no other method for mapping light and color features (regardless of the image contents). Therefore, our Light to Colors policy offers the first solution to that kind of mapping problem.

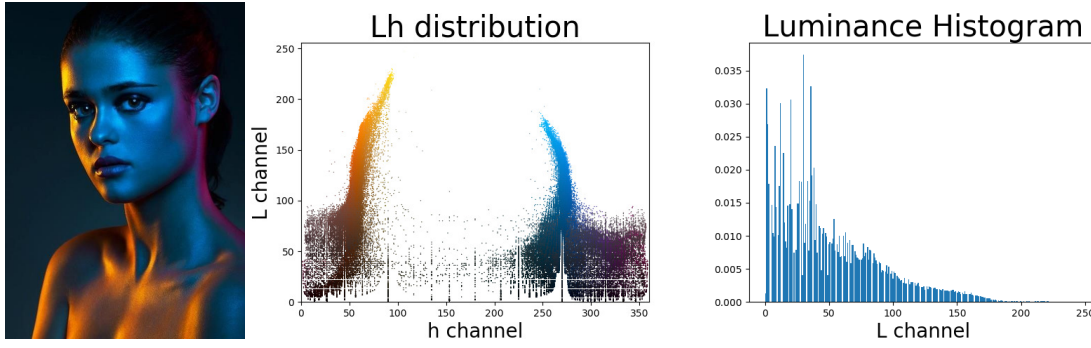
### 3.3.2 Colors to Light

Now, we consider a colors-based style input image and a light-based style target image. A policy that maps light features to color features has not been tackled by existing methods so far. We propose a strategy similar to the Light to Colors policy. The Colors to Light policy maps the highlight areas in the target image to the warmest color in the input image and vice versa, the darkest areas in the target image correspond to the coldest color in the input image.

### 3.3.3 Light to Light

The mapping between two light-based style images is handled by Bonneel’s luminance-based mapping [6]. Three luminance clusters are considered, namely shadows, midtones and highlights. There are images for which one of the clusters is insignificant. In that





**Figure 5.7** – An artistic image, its Luminance-Hue distribution and its luminance histogram. The luminance histogram (on the right-hand side) shows that the image is overall dark and it does not contain highly contrasting regions due to lack of highlights. Even though highlights are not present in the image (quantitatively speaking), they are visually perceived thanks to the carefully chosen image colors. The photographer has used three color clusters, *i.e.* orange-yellow, blue and purple, which appear consequently on the girl’s face and neck from left to right (and are also illustrated in the Luminance-Hue distribution in the same order). These three color clusters indicate highlights, soft and hard shadows respectively. Our Colors to Light and Light to Colors mapping policies are based on this artistic approach.

case, only two luminance clusters are considered. Light to Light policy maps shadows to shadows, midtones to midtones and highlights to highlights.

### 3.3.4 Colors to Colors

The last mapping policy maps the clusters of two colors-based style images. There are several logical solutions to that kind of problem. For instance, Bonneel et al. [6] carry out a luminance-based mapping for such kind of transfer. However, when the input and target images are colors-based style images, Bonneel’s method does not produce plausible results because it does not consider color clusters in the mapping process. Another possible solution involves mapping together the most similar clusters in terms of average hue. However, such an approach would not prevent the input clusters from being mapped to only one of the target clusters. Likewise, one could also sort the input and target clusters by their hue values and map the corresponding clusters together. In contrast, we believe that users would expect similar colors to be mapped to similar colors which is not guaranteed by the latter mapping approach. We have experimented with these three mapping solutions and we came up with a more general solution, as described hereafter.

At each iteration of algorithm 1, we map the two most similar input and target clusters, *i.e.* those with the minimum Euclidean color distance between their centers. Put differently, the centers of the most similar clusters are close in terms of hue. The function  $\text{FindMinDistPair}(\zeta_{lch}^I(h), \zeta_{lch}^J(h))$  aims to find the most similar clusters  $\mathbf{I}_{lab}^{(s_1)}$  and  $\mathbf{J}_{lab}^{(s_1)}$  and their probability functions  $f_{lab}^{(s_1)}$  and  $g_{lab}^{(s_1)}$  at each iteration of algorithm 1. The strategy ensures one-to-one mapping where the nearest colors are associated first. Like in the previous two policies, the corresponding probability functions are passed to the function  $\text{PerformTranspOnAB}()$ , handling the color transformation<sup>2</sup>. Finally, we exclude elements

<sup>2</sup>Whatever the used policy, the color transformation is always performed on the chroma axes ( $a$  and  $b$ )

from the sets  $\zeta_{lch}^{\mathbf{I}}(h)$  and  $\zeta_{lch}^{\mathbf{J}}(h)$  at the end of each step. That way, the final iteration of the algorithm maps the two most distant colors.

### 3.4 Color transfer method

Once we map the target to the input clusters, a color transfer is performed between each pair of corresponding clusters. Our color transfer consists of a color transformation and a local chromatic adaptation. The color transformation is handled by the function  $PerformTranspOnAB(f_{lab}^{(s_I)}, g_{lab}^{(s_J)})$  in algorithm 1. We carry out the color transformation in the CIE Lab color space. We separate the luminance channel from the chroma channels. Indeed, the human eye is much more sensitive to changes in the light conditions than to changes in the colors. Therefore, we apply the color transformation only on the chroma channels (whatever the determined policy). Finally, we use local chromatic adaptation transform (CAT), handled by the function  $CATLocal(\mathbf{O}_{rgb}, \mathbf{J}_{rgb})$  in algorithm 1, to reproduce the lighting conditions of the target image. These steps are discussed in the following subsections.

#### 3.4.1 Color transformation on the chroma channels

The clustering step of our method performs a partitioning of the input and target images into homogeneous clusters in the Lab color space. Their 2D  $ab$  distributions can be modeled by 2D Gaussian distributions. Therefore, a parametric color transfer approach is used for carrying out the color transfer between the cluster  $ab$  distributions.

We adopted the parametric color transfer method, proposed by Pitié et al. [60]. Pitié’s method builds a mapping  $t(\mathbf{I})$  between the image input  $\mathbf{I}$  and the target image  $\mathbf{J}$ . The mapping  $t(\mathbf{I})$  transforms the input distribution  $f(\mathbf{I})$  into a distribution similar to the target distribution  $g(\mathbf{J})$ . To build the mapping  $t(\mathbf{I})$ , Pitié et al. assume that  $f(\mathbf{I})$  and  $g(\mathbf{J})$  follow a multivariate Gaussian law. For each corresponding pair of input and target Gaussian clusters  $\mathbf{I}_{lab}^{(k)}$  and  $\mathbf{J}_{lab}^{(k)}$ , we build a mapping  $t_k(\mathbf{I}_{lab}^{(k)})$  consistent with the proposed mapping by Pitié et al. [60], *i.e.* derived as a closed-form solution [60–62] to an optimal transportation problem well-known as the Monge-Kantorovich optimization problem [63]. That way, for each pair of clusters, we build a unique mapping which minimizes the overall cost of the color transfer [60]. More details and discussion about the Monge-Kantorovich transformation are presented in chapter 6.

#### 3.4.2 Overlapping

When using a clustering technique, we need to take care of the strong color difference which may occur between the clusters. To achieve a smooth transition between the clusters and to lessen the visibility of eventual artifacts caused by the color transformation, we let the input clusters overlap around their spatial boundaries. Associating pixels with more than one cluster is known as fuzzy (soft) clustering [64, 65]. Each pixel  $i$  is assigned

---

of the 3D distributions  $f_{lab}^{(s_I)}$  and  $g_{lab}^{(s_J)}$ .

a probability  $p_{ik}$  to belong to a cluster  $k$  as follows:

$$p_{ik} = \alpha_{ik} / \sum_{j=1}^N \alpha_{ij} \quad (5.3)$$

where  $N$  is the number of clusters and  $\sum_{j=1}^N p_{ij} = 1$ . Additionally,  $\alpha_{ik}$  is defined as  $\alpha_{ik} = \exp(-D_M^2(\mathbf{x}_i, f_{CS}^{(k)}))$ , where  $D_M(x_i, f_{CS}^{(k)})$  is the Mahalanobis distance [66] for a 3D vector  $\mathbf{x}_i$  with values in the Lab color space for the  $i^{th}$  pixel. The Mahalanobis distance measures the distance of each overlapping pixel to one of the Gaussian cluster distributions in our model. Finally, the values of the  $a$  and  $b$  channels for the output image  $\mathbf{O}_{lab}$  are computed as follows:

$$\mathbf{O}_i = \sum_{j=1}^N p_{ij} \mathbf{t}_{ij} \quad (5.4)$$

where  $\mathbf{t}_{ij}$  is the vector of chroma values for the  $i^{th}$  pixel, obtained from the transformation for the  $j^{th}$  cluster. The output  $\mathbf{O}_i$  is the vector of the transformed chroma values for the  $i^{th}$  pixel. Results, obtained after the color transformation in our method, as shown in figures 5.8, 5.9 and 5.13.

### 3.4.3 Local chromatic adaptation

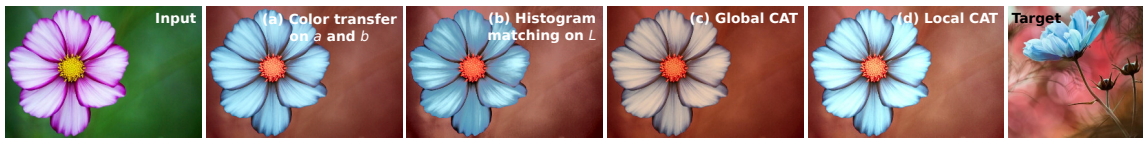
The color transformation, described in the previous section, is applied only on the chroma channels of the input image. Therefore, to complete the color transfer, we need to reproduce the light of the target image. To perform this task, Bonneel et al. [6] apply a naive histogram matching on the luminance channel of CIE Lab. The naive histogram matching for the luminance channel may cause artifacts and highly saturated results for images with very different lighting set-ups. This makes the naive histogram matching unsuitable for our purposes, as shown in figure 5.8 (result (b)).

As a final stage of our color transfer method, we apply a local CAT algorithm on the image  $\mathbf{O}_{rgb}$ , obtained with the Monge-Kantorovich color transformation. Local CAT aims to adapt pixel-wise the colors of image  $\mathbf{O}_{rgb}$  to the target illuminant. This way, undesired color saturation is avoided and naturalism is preserved. Similarly to the iCAM algorithm [21], we apply CAT locally to the pixels of the input image by computing a “white image” using a Gaussian low-pass filter with a kernel size equal to half the sum of the image dimensions. Each input pixel is influenced by the chromatic transform and therefore, local luminance variations are captured efficiently and reproduced in the result. Indeed, the local CAT enhances the contrast and prevents the image from becoming flat (refer to examples 3 and 5 in Figure 5.9).

Frigo et al. [20] were the first to apply the CAT algorithm iteratively. Instead of adapting the colors of an image to a well-known illuminant, they have used a global estimation of the input illuminant and the target illuminant by assuming Gray World [18]. Likewise, we estimate the target illuminant as the average of the non-gray pixels in the target image [18,20]. Unlike [20], we apply the local CAT only once and not in an iterative manner, which results in a decrease of the computational time. As shown in Figure 5.8, our local CAT adapts the colors of image  $\mathbf{O}_{rgb}$  to the target illuminant better than the global CAT [20]. Therefore, the local CAT manages to reproduce better the colors of the target

image. Figure 5.8 also illustrates the impact of the local CAT on the final result. The optimal color transformation of our method changes only the colors of the input image so that they become similar to the target colors. However, as we exclude the luminance channel from the transformation, the luminance of image  $\mathbf{O}_{rgb}$  remains relatively unchanged. As shown in figure 5.8, once we locally adjust the colors of image  $\mathbf{O}_{rgb}$  to the target illuminant, we improve the similarity between our result and the target color palette. The influence of the local CAT on the colors of the final result is also shown in figures 5.9 and 5.13.

For all results, shown in this chapter, the local CAT has been applied in the LMS color space (similarly to the CAT algorithm, presented in part 2). The adaptation factor in the local CAT was scaled by 0.3 [21] and the value of the surround factor was set to 1.



**Figure 5.8** – Comparison between the naive histogram matching, the global CAT and the local CAT. Result (a) is obtained after the color transformation for the chroma channels  $a$  and  $b$  in our method. Result (b) is obtained by applying a histogram matching on the luminance channel of result (a). Despite the color similarity between result (b) and the target image, result (b) is highly saturated and unnatural. Result (c) is obtained iteratively by applying a global CAT [20] on image  $\mathbf{O}_{rgb}$ , whereas result (d) is obtained using only one iteration of our local CAT. This figure also presents the impact of the local CAT on the final result. Our color transformation modifies only the chroma channels of CIE Lab and keeps the input luminance relatively unchanged, as shown in result (a). The local CAT further improves the similarity between result (d) and the target color palette (note the background color and the better illuminated flower petals in result (d)).

### 3.5 Implementation details

The implementation of our algorithm begins with the classification of images into one of two classes. Our classification system depends on three parameters with default values as follows:  $c_{min} = 10$ ,  $d_p = 30$ ,  $s_{min} = 0.05 \times n$ , where  $n$  is the number of pixels in the image. The parameters remain fixed throughout the computation. The default values were determined after several experiments.

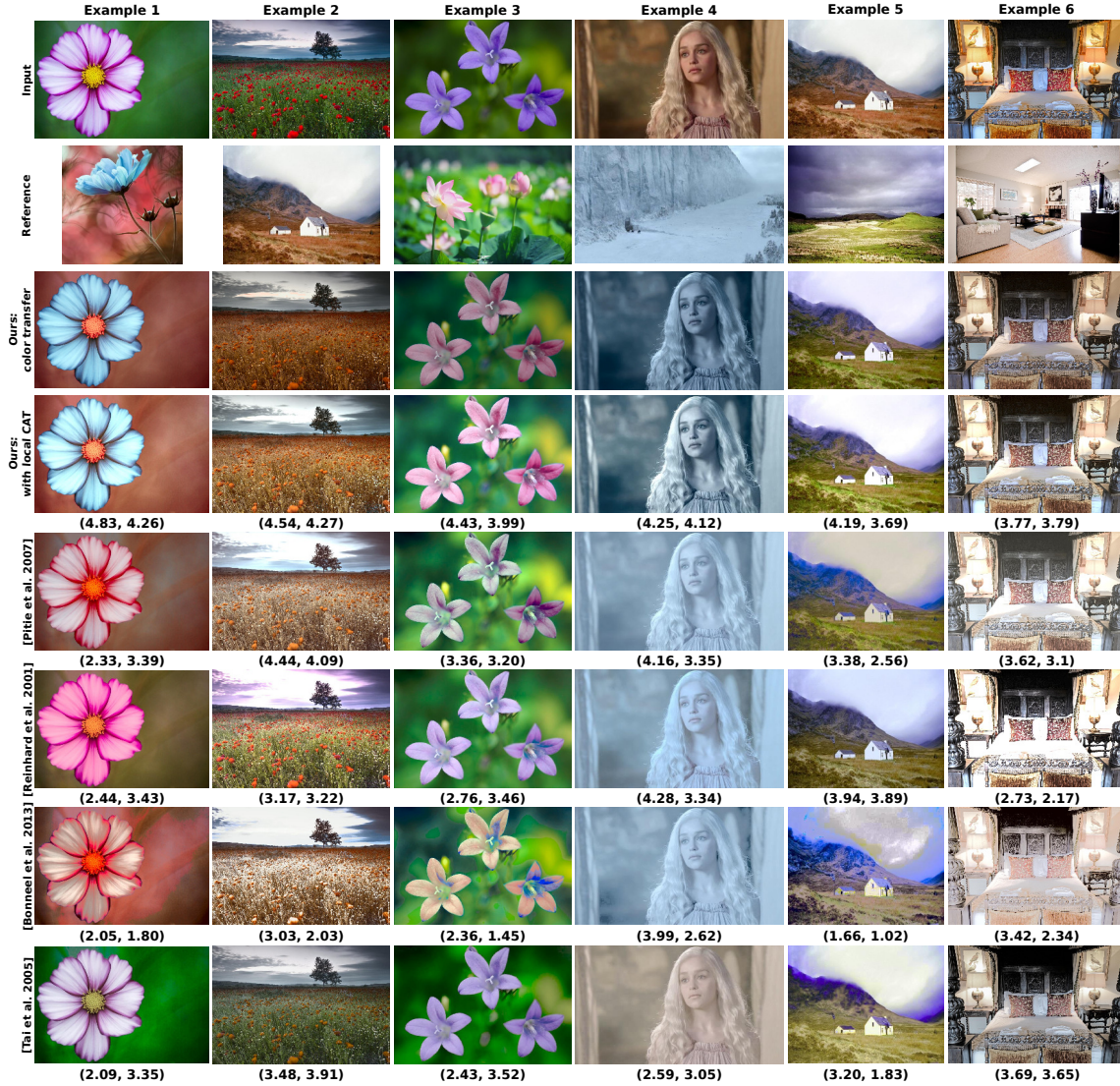
Furthermore, the image classification and clustering are handled in the Lch color space. At the same time, we use the Lab color space to carry out the color transfer. We have implemented our algorithm in C++. The proposed algorithm has been performed on a laptop with an Intel Core i7 2.10GHz and 16Go RAM. For an image of 1000x1000 pixels, the average execution time is 25s (without optimization).

## 4 Results and evaluation

We compared results, obtained with the proposed method, to results, obtained with four other methods. On one hand, we chose two state-of-the-art global transformations by



Reinhard et al. [3] and Pitié et al. [60]. That way, we show that, often, global transformations do not preserve the photo-realism of the input image but produce rather saturated results. On the other hand, we chose Bonneel et al.’s and Tai et al.’s local color transfer methods [5, 6] to demonstrate that using a luminance-based cluster mapping (regardless of the type of images) does not always ensure a good color transfer.



**Figure 5.9** – Results from our method and four state-of-the-art methods. The style and the aesthetic scores are presented below each result (in the order: (style, aesthetics)). The optimal value for both scores is 5. Our method obtains the highest aesthetic scores because it produces natural results with perceptually pleasing contrast. The participants in our user study have also given our results the highest style scores. An exception is example 4, for which Reinhard et al.’s result obtains a slightly higher style score than the style score for our result.

Figure 5.9 shows several results. As observed from examples 1 and 3 in figure 5.9, the proposed method manages to transfer properly the colors of the foreground and the background without the need of segmentation [6] or saliency [20]. Most of our results respect the semantic of the input image and the style of the target image. For instance, the input and target images in examples 2 and 5 respectively are close in terms of image content.

We use our Colors to Colors mapping policy to map together the similar input/target areas (*e.g.* sky is mapped to sky and ground is mapped to ground). Colors to Colors mapping policy involves only the input and target color distributions (as opposed to texture-based mappings [67]). Furthermore, example 4 in figure 5.9 illustrates the efficiency of our method when applied to images with significantly different contents. Despite the difference in contents between the input and target images, our method obtains a naturally pleasing final result. Thanks to the local CAT in our method over-/under-saturated images are unlikely to be produced by the proposed method (examples 2, 4, 6). Figure 5.13 presents four additional results.

## 4.1 User study evaluation

To compare the five methods in terms of style transfer and visual pleasingness of the results, we conducted a subjective evaluation study. We asked 15 users to evaluate 50 results obtained for 10 input and 10 target images for the five methods. The input and target images vary in content, semantics, lighting set-up, color features. The input and target images in our user study were selected from the various photographic collections of *500pixels.com*<sup>3</sup>. The input, target and resulting images used as test cases in our subjective evaluation are shown in figures 5.9 and 5.13.

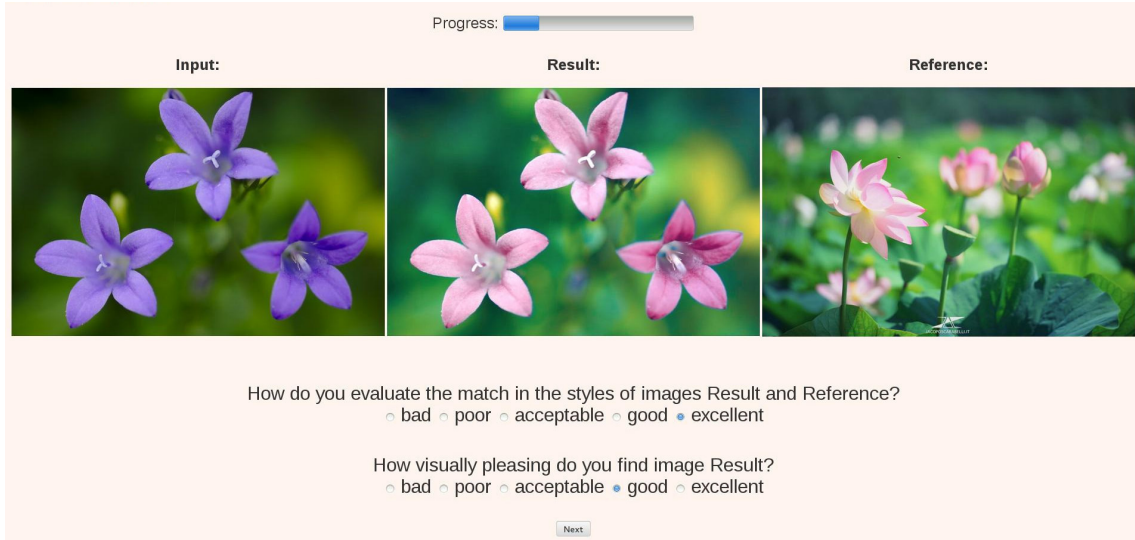
The participants were 23 to 53 years old. The majority of them had average image editing expertise. Five of them wore glasses and none of them suffered from a color vision deficiency. Each participant was presented with triples of images consisting of an input image, a target image and a result, obtained from a color transfer between the input and target images. Moreover, we added a special image triple, referred to as a baseline, to the set of the 50 image triples. The baseline was the only triple for which the colors of the result and the target image differed significantly one from another. Therefore, users were expected to give the baseline the lowest score. That way we tested users' judgments on the rest of the results.

First, the participants were asked to evaluate the match in styles (in terms of colors and light) between each result and its corresponding target image. The perceived match in styles was closely related to users' expectations. Second, the users were asked to evaluate how visually pleasing the results were. The evaluation of the visual pleasingness was based on users' perception of the aesthetics of the results. Five-point scale (5-excellent, 4-good, 3-acceptable, 2-poor, 1-bad) was used to evaluate the results for both tasks. Moreover, four repetitions per result were used to minimize any possible bias and to increase the robustness. Figure 5.10 shows a screen shot of the system we designed to conduct the user study.

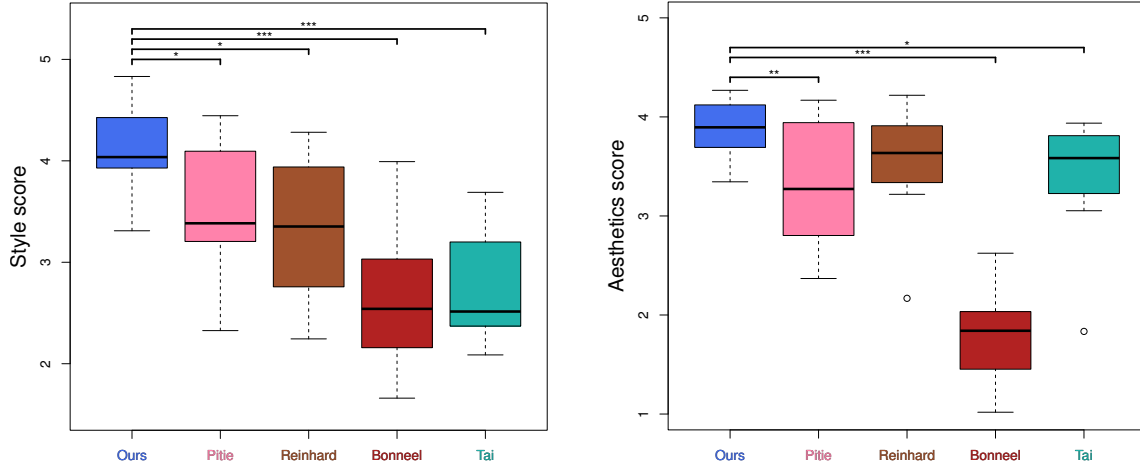
Each user evaluated the results individually. The indoor conditions, the display properties and the relative proximity of the user to the display were the same. In contrast, the order of displaying the image triples was random and different for each participant. Finally, a short training session took place before the real evaluation during which the users adapted to the tasks of the real test.

The scores, obtained for each of the four repetitions, were combined into a single score per result. This score is computed as 1) the mean, 2) the weighted mean (giving different

<sup>3</sup>The input and target images from Example 5 in figure 5.9 were borrowed from Pitié et al.'s paper [60].



**Figure 5.10** – Screen shot of the system we designed to conduct the user study.



**Figure 5.11** – Box-and-Whiskers plots per method, displaying the distributions of the style and aesthetics scores. The connecting brackets indicate significant difference between our method and the corresponding state-of-the-art method ("\*" stands for  $p\text{-value} \leq 0.05$ , "\*\*\*" stands for  $p\text{-value} \leq 0.01$  and "\*\*\*\*" stands for  $p\text{-value} \leq 0.001$ ).

weights to each repetition) and 3) the median. Paired t-tests have shown that there is no significant difference between the three statistical approaches. Therefore, we carry out the analysis using the mean scores. To obtain the final score per result, we normalize the mean score by the baseline score.

We refer to the scores, corresponding to the first and second questions in our user study as the style and aesthetic scores respectively. Figure 5.11 displays the Box-and-Whiskers plots of the score distribution for each of the two types of scores and for each of the five color transfer methods. As observed, our method has the highest mean style score and the highest aesthetic score. Moreover, the style and aesthetic score deviations of our method are significantly smaller than the style and aesthetic deviations of the other four methods.

Furthermore, we performed paired t-tests between the score distribution of our method and the score distributions of the other four methods (for both the style and aesthetic scores). The paired t-tests have shown a significant difference in the style scores between



our method and each of the other four color transfer methods. Furthermore, the tests have indicated that the aesthetic scores of our method differ significantly from the aesthetic scores of three of the state-of-the-art methods [5, 6, 42]. In contrast, there is no significant difference between the aesthetic scores of the proposed method and the aesthetic scores of Reinhard et al. method [3]. Both methods obtain high aesthetic scores. To this end, both of them tend to produce visually pleasing images, as seen from the examples, shown in this chapter.

As shown in table 5.2, the style and aesthetic scores for our method are highly correlated, indicating that the proposed method produces consistently good results in terms of style transfer and visual pleasingness of the results.

## 4.2 Objective metric evaluation

In addition to the subjective user evaluation, an objective evaluation of the results was carried out. A good color transfer has to ensure artifact-free images as well as to properly transfer the light and color distributions of the target images. We believe that both criteria are equally important. Therefore, for the objective evaluation we use two objective metrics.

As depicted in [20, 44], the metric **SSIM** [68] can be used to measure the degree of artifacts in the result. The SSIM is a function of three comparison components, *i.e.* luminance  $l$ , contrast  $c$  and structure  $s$ , computed as follows:

$$SSIM(\mathbf{x}, \mathbf{y}) = f(l(\mathbf{x}, \mathbf{y}), c(\mathbf{x}, \mathbf{y}), s(\mathbf{x}, \mathbf{y})) \quad (5.5)$$

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + c_1}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + c_1} \quad (5.6)$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + c_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + c_2} \quad (5.7)$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}} + c_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}} + c_3}, \quad (5.8)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  denote corresponding patches from two gray-scale images,  $\mu_{\mathbf{x}}$  and  $\sigma_{\mathbf{x}}$  ( $\mu_{\mathbf{y}}$  and  $\sigma_{\mathbf{y}}$ ) are the mean and the standard deviation in the patch  $\mathbf{x}$  ( $\mathbf{y}$ ),  $\sigma_{\mathbf{xy}}$  denotes the standard deviation between the patches  $\mathbf{x}$  and  $\mathbf{y}$ , and  $c_1$ ,  $c_2$  and  $c_3$  are stabilizing variables.

We apply the SSIM between the luminance channels of the input image and the result (similarly to [20]). However, as the goal of a color transfer method is to transform the input luminance so that it gets similar to the target luminance, the input luminance will differ from the luminance of the result. That is why, unlike the methods in [20, 44], we removed the luminance component  $l$  from the computation of the metric (equation 5.5) and proceeded with only the contrast and structural components of the SSIM.

The **Bhattacharya coefficient** is used to measure the distance between two histograms  $\mathbf{x}$  and  $\mathbf{y}$  [69, 70] as follows:

$$BC(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^n \sqrt{\mathbf{x}_i \mathbf{y}_i} \quad (5.9)$$

where  $\mathbf{x}_i$  and  $\mathbf{y}_i$  are samples from the two histograms. To evaluate how successful the color transfer is, we apply it on the color and luminance histograms of the result and the target image (we use the Lab color space). We compute a single Bhattacharya coefficient score

Method	Ours	Pitie	Reinhard	Bonneel	Tai
Correlation	0.86	0.49	0.50	0.93	-0.05

**Table 5.2** – Correlations between the style and aesthetics scores.

Metrics / Mean values	Ours	Pitie	Reinhard	Bonneel	Tai
SSIM	0.98	0.97	0.95	0.89	0.96
Bhattacharya	0.86	0.88	0.85	0.92	0.73

**Table 5.3** – Mean values of the SSIM and the Bhattacharya coefficient for each of five methods.

by averaging the Bhattacharya coefficient scores, obtained for the luminance channel and the two chroma channels.

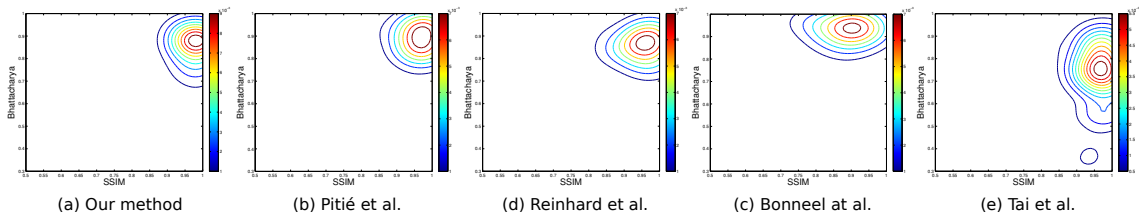
The pair of values (1, 1) is optimal for the pair of metric scores (SSIM, Bhattacharya). It refers to results with the same visual quality as that of the input image and with exactly the same light and color distributions as those of the target image.

For each of the five color transfer methods and for each image in a set of 40 image results per method (10 of which were also used in the user study), we plot the pair of values (SSIM, Bhattacharya coefficient). Contour plots are used to illustrate the joint density of the SSIM and the Bhattacharya coefficient for each method, as shown in figure 5.12. Several observations can be made.

First, we observe that our approach obtains the highest average SSIM value, as presented in table 5.3. Moreover, paired t-tests have shown that SSIM values, obtained with our method, differ significantly from those obtained with the other four methods. Therefore, the proposed method is the best one among the five methods when it comes to producing artifact-free final images.

Furthermore, the centers of the contour plots for our method and Pitié’s method are concentrated around the optimal value (SSIM, Bhattacharya) = (1, 1). The same can be observed for the methods in [3, 6]. Although Bonneel et al. [6] obtain the highest mean value for the Bhattacharya coefficient, their method is likely to cause significant number of artifacts to the final images. On the other hand, the two objective metrics show that Tai et al.’s method [5] is expected to produce less artifacts but it is also less efficient in terms of color transfer. Finally, there is no significant difference between our method and both Pitié’s and Reinhard’s methods regarding the Bhattacharya coefficient.

To conclude, the proposed method succeeds in transferring the color and light distributions of the target image with respect to the target style while preserving the naturalism in the results and keeping the degree of artifacts low.

**Figure 5.12** – Contour plots for the set of metric pairs (SSIM, Bhattacharya) for each of five method. The majority of images has values which lie in the small areas displayed in red.

## 5 Limitations

The results as well as the subjective and objective evaluations, presented in this chapter, have demonstrated that our style-aware method outperforms results obtained with state-of-the-art color transfer methods. Nevertheless, our method has also some limitations. Example 10 in figure 5.13 shows one of them. According to the participants of our user study, our result in example 10 is far from their expectations. Users expected that the orange color of the buildings in the target image would be transferred to the buildings in the input image. Therefore, our result in example 10 got the lowest score image among all the images, obtained with our method. The state-of-the-art methods also fail to properly color the buildings in the input image. This challenging case could be solved by using additional constraints such as saliency [20], content-based transfer [71], and texture-guided transfer [67].

## 6 Conclusion

This chapter focused on developing a new way for style transfer for a wide class of image pairs. We introduced a cluster-based style transfer method which outperforms state-of-the-art approaches. Furthermore, we developed an automatic way of feature detection in images for the two image characteristics: light and colors. Our most important contribution lies in the development of three mapping policies which make it possible to carry out a local color transfer between various pairs of images regardless of their content. The designed policies can be applied independently from the color transformation between the input and target images.

Thanks to our four mapping policies, our results manage to portray well the target style. In this chapter, we defined the style of images as a combination of color and light image features. However, the image style may include a number of other features, such gradient, contrast, texture, etc. To extent the color transfer to a feature transfer between images, we need a general model, which describes the distributions of all considered features. The quality of our color transfer depends on how well the MGD fits the given data. To this end, the color transformation, used in our style-aware color transfer method, may not be appropriate for a feature transfer between images, as the MGD fails to fit well heavy-tailed distribution, *e.g.* the distributions of image gradient fields. In the following chapter, we present a transformation of a new class of distributions, called the multivariate generalized Gaussian distribution. This class includes both heavy-tailed and light-tailed distribution making it suitable for fitting the distributions of various image features, such as color, gradient, texture wavelets, etc. Using the multivariate generalized Gaussian distribution, we carry out a multi-dimensional feature transfer between images, as described in the following chapter.





**Figure 5.13** – Results from our method and four state-of-the-art methods, used in our user study. The style and the aesthetic scores are presented below each result (in the order: (style, aesthetics)). Our results, shown in this figure, obtain fairly high style and aesthetic scores compared to Pitié et al.’s and Reinhard et al.’s results. Users agree that our results are artifact-free, aesthetically pleasing and naturally looking images. Example 10 is an extremely challenging test case for our method as well as for all four reference methods, and illustrates a limitation of our method.

# Transformation of the Multivariate Generalized Gaussian Distribution for Image Editing 6

---

## 1 Introduction

Multivariate Gaussian distribution (MGD) is commonly used in image processing applications to fit the distributions of image features [34]. Thanks to the analytically tractable density function of the MGD, there exists a number of MGD-based statistical transformations, applied to carry out a color transfer between images [34]. As shown in the previous chapter, such transformations benefit from the analytical properties of the MGD, but they also depend on how accurately the MGD approximates the given distributions. To this end, such transformations fail when applied to heavy-tailed distributions (*e.g.* sparse distributions with exponentially unbounded tails [72]) such as the distributions of image gradient fields and wavelet coefficients. To overcome this limitation, the multivariate Laplacian distribution (MLD) is usually adopted to fit sparse distributions. The MLD is often used in speech recognition to characterize discrete Fourier coefficients [37] as well as in image processing to model gradient fields [36].

As the MGD and the MLD are special cases of the multivariate generalized Gaussian distribution (MGGD) with shape parameters equal to 1 and 0.5 respectively, the accuracy of the statistical approximation could be improved by relaxing the constraint on the shape parameters. The MGGD and its properties have been introduced in [22, 73]. The property of the MGGD to model accurately sparse distributions (for a shape parameter less than 1) has been exploited in many applications, such as texture discrimination and texture retrieval [30, 32, 74], image and video segmentation [75], image de-noising [28, 76]. Including the MGD and the MLD as special cases, the MGGD with an unconstrained shape parameter is likely to accurately fit a wide class of image feature distributions, including those of color, gradient, wavelet coefficients, etc.

Despite the major role of the MGGD in image processing, no transformation between two MGGDs has so far been tackled. To address this limitation, we propose a novel transformation between input and target sets of sample vectors, following an MGGD with given parameters. The transformation consists of two main steps. First, a linear optimization transforms the input sample vectors so that their distribution approximates the second-order statistics of the target distribution. Second, a stochastic-based transfer modifies the sample vectors, computed during the first step, so that their distribution gets similar to the target distribution (in terms of both scale and shape). We demonstrate the potential of our method for color and gradient transfers between images. The proposed



transformation allows to simultaneously transfer a number of image features by taking into account the dependencies between them. Unlike existing image editing techniques, which are limited to only color transformations, we demonstrate the efficiency of our method for a  $p$ -dimensional feature transfer between images.

To sum up, the main contributions of the method, presented in this chapter, are twofold:

- a novel transformation between MGGDs, consisting of a linear Monge-Kantorovich transformation and a stochastic-based shape parameter transfer;
- two novel applications, carried out by the proposed MGGD-based transformation:
  - a transfer of gradient between images;
  - a multidimensional transfer of color and gradient between images.

The chapter is organized as follows. Section 2 presents related work on existing transformations between MGDs as well as applications of these transformations. Section 2 introduces our transformation between MGGDs and provides details about the evaluation of the transformation. Applications and results are presented in section 4. Finally, the last section concludes the chapter.

## 2 Related work on MGD transformations

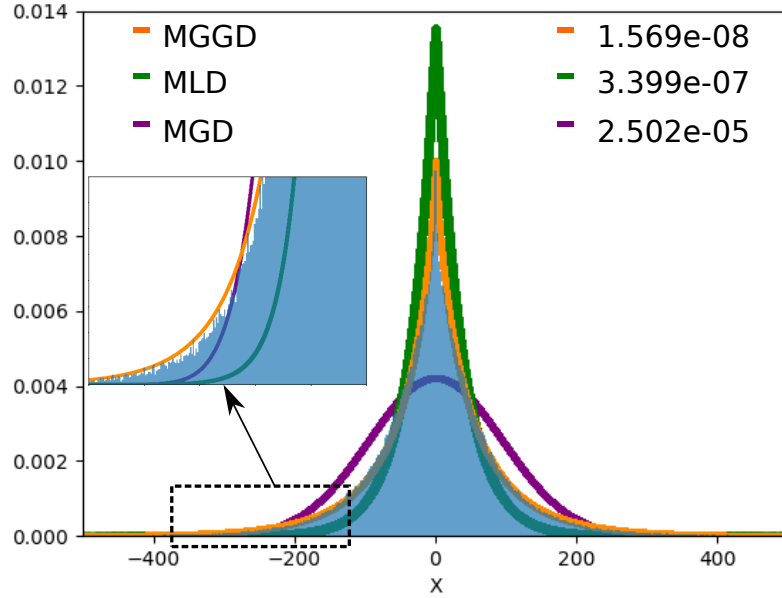
As this chapter tackles transformations between MGGDs, the following sections discuss a well-known special case of the MGGD - the MGD. Hereafter, we briefly present related work, connected to existing MGD transformations.

Several linear transformations of the MGD have already been presented in chapter 5 (section 2), *i.e.* the square-root decomposition, the Cholesky decomposition and the Monge-Kantorovich transformation. The MGD describes well the distributions of color in images [34] and therefore, these MGD-based transformations are often employed in the context of image and video color and transfer [6, 34], as discussed in the previous chapter.

The MGD-based transformations also play a role in applications, such as color correction for image mosaicking (stitching). Before stitching two images, one needs to make sure that their colors match (which is hardly the case due to different light conditions, camera settings, white balance, etc.). Several parametric models for color correction, based on Reinhard *et al.*'s global parametric method [3], have been proposed. Oliveira *et al.* [2] carry out a weighted version of Reinhard's transformation for coarse registered images. Tai *et al.* [5] and Xiang *et al.* [77] apply probabilistic image segmentation using Gaussian mixture models and then carry out Reinhard's transformation between corresponding regions.

Despite the importance of the MGD-based transformations, their applicability is limited, as the MGD is not appropriate for modeling heavy-tailed distributions, as illustrated in figure 6.1. To this end, sparse distributions are often fitted by the MLD. The MLD has been adopted in the context of image denoising for modeling the distributions of Lapped transform coefficients and wavelet coefficients as well as for learning non-linear dependencies between natural images [78–80].

Both the MGD and the MLD belong to the family of the MGGD with shape parameters equal to 1 and 0.5 respectively. If we relax the constraint on the shape parameter, we can



**Figure 6.1** – Comparison between the MGD, the MLD and the MGGD for  $p = 1$  (univariate case). The histogram (in blue) shows a heavy-tailed distribution of real-world data. The mean and the standard deviation of the MGD are calculated from the data and are equal to 0 and 95 respectively. The shape and scale parameters of the MGD are equal to 0.5 and 341 respectively, whereas those of the MGGD are equal to 0.39 and 341 respectively. Unlike the MGD, both the MLD and the MGGD depict the sparseness of the original distribution. The MGGD models the heavy-tails of the original distribution more accurately than both the MLD and the MGD. The accuracy of the fit for each approximation is given by the mean square error (top-right corner). The MGGD with a shape parameter equal to 0.39 proves to be the best fit for the real-world data.

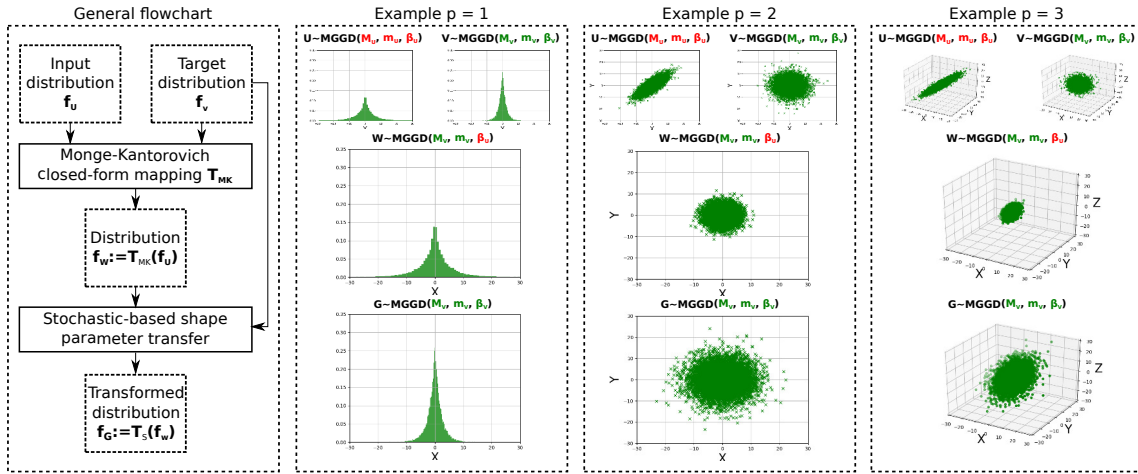
describe a wide class of  $p$ -dimensional features (such as color, gradient, wavelets) by only one probabilistic model. To make the transform between such probabilistic models possible, we propose a novel transformation between MGGDs as presented hereafter.

### 3 Transformation of the MGGD

In the following sections, we introduce the main contribution of this chapter: a transformation between MGGDs. First, we present the parameters of MGGD and a method for computing their maximum likelihood estimators. Second, we introduce our MGGD-based transformation. As a remark, the density function of the MGGD is already introduced in part 2.

#### 3.1 Multivariate generalized Gaussian distribution

**MGGD parameter estimation.** Let  $\mathbf{x} \in \mathbb{R}^p$  be a random vector which follows the law of an MGGD with unknown parameters  $\mathbf{M}$ ,  $m$  and  $\beta$  and a **zero mean**. Let  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  be



**Figure 6.2** – The general flowchart of the proposed transformation is presented in the left box, whereas the other three boxes contain examples of 1D ( $p = 1$ ), 2D ( $p = 2$ ) and 3D ( $p = 3$ ) distribution plots respectively ( $p$  denotes the dimension). Each of the example boxes presents plots of the distribution  $f_U$  of the input set  $U$ , the distribution  $f_V$  of the target set  $V$  and the transformed distribution at each step of the flowchart (of sets  $W$  and  $G$ ) in the corresponding dimension (best viewed on screen).

sample vectors of the random vector  $\mathbf{x}$ . The MLEs of the unknown parameters  $\mathbf{M}$ ,  $m$  and  $\beta$  can be obtained by differentiating the log-likelihood of the sample vectors  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  with respect to  $m$ ,  $\mathbf{M}$  and  $\beta$ , yielding the following three equations [24]:

$$m = \left( \frac{\beta}{pN} \sum_{i=1}^N (\mathbf{x}_i^T \mathbf{M}^{-1} \mathbf{x}_i)^\beta \right)^{\frac{1}{\beta}}, \quad (6.1)$$

$$\mathbf{M} = \sum_{i=1}^N \frac{p}{\mathbf{y}_i + \mathbf{y}_i^{1-\beta} \sum_{j \neq i} \mathbf{y}_j^\beta} \mathbf{x}_i \mathbf{x}_i^T, \quad (6.2)$$

$$\begin{aligned} \alpha(\beta) = & \frac{pN}{2 \sum_{i=1}^N \mathbf{y}_i^\beta} \sum_{i=1}^N (\mathbf{y}_i^\beta \ln(\mathbf{y}_i)) - \frac{pN}{2\beta} \ln \left( \frac{\beta}{pN} \sum_{i=1}^N \mathbf{y}_i^\beta \right) \\ & - N - \frac{pN}{2\beta} \left( \psi \left( \frac{p}{2\beta} \right) + \ln 2 \right) = 0, \end{aligned} \quad (6.3)$$

where  $\mathbf{y}_i = \mathbf{x}_i^T \mathbf{M}^{-1} \mathbf{x}_i$  and  $\psi(\cdot)$  is the digamma function [81].

Equations (6.2) and (6.3) indicate that the matrix  $\mathbf{M}$  and the shape parameter  $\beta$  do not depend on the scale parameter  $m$  and can be alternatively estimated. As discussed in [24, 82], a recursive algorithm, associated to equations (6.2) and (6.3), is used to compute the MLEs of  $\mathbf{M}$  and  $\beta$ . During each iteration  $k$  of the algorithm, the Newton-Raphson procedure is carried out to approximate the exact solution of equation (6.3), where  $\mathbf{M}$  is replaced by the matrix  $\mathbf{M}^{(k)}$ , estimated during iteration  $k$ . Once the recursive algorithm converges, the scale parameter  $m$  is estimated using equation (6.1), where  $\mathbf{M}$  and  $\beta$  are replaced by their MLEs. The MLEs of the parameters  $\mathbf{M}$ ,  $m$  and  $\beta$  are computed for values of  $\beta$  in the interval  $(0, 1]$ , as discussed in [24].

## 3.2 Transformation between two MGGDs

In the present section we introduce our transformation between two MGGDs. Let  $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_N)$ ,  $\mathbf{u}_i \in \mathbb{R}^p$  for  $i \in \{1, \dots, N\}$ , and  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_L)$ ,  $\mathbf{v}_j \in \mathbb{R}^p$  for  $j \in \{1, \dots, L\}$ , be the input and target sets of sample vectors of random vectors  $\mathbf{u}$  and  $\mathbf{v}$ , distributed according to an MGGD with a zero mean:  $\mathbf{U} \sim \text{MGGD}(\mathbf{M}_{\mathbf{U}}, m_{\mathbf{U}}, \beta_{\mathbf{U}})$  and  $\mathbf{V} \sim \text{MGGD}(\mathbf{M}_{\mathbf{V}}, m_{\mathbf{V}}, \beta_{\mathbf{V}})$ . In practice, the sets  $\mathbf{U}$  and  $\mathbf{V}$  can contain RGB image pixel values, image gradient pixels, etc. The distributions of  $\mathbf{U}$  and  $\mathbf{V}$  are denoted by  $f_{\mathbf{U}}$  and  $f_{\mathbf{V}}$  respectively. The parameters of  $f_{\mathbf{U}}$  and  $f_{\mathbf{V}}$  are obtained by applying the recursive algorithm, associated to equations (6.1), (6.2) and (6.3) (for  $\beta_{\mathbf{U}}, \beta_{\mathbf{V}} \in (0, 1]$ ).

The goal of the transformation between MGGDs is to transform the input set of sample vectors  $\mathbf{U}$  into an output set of sample vectors  $\mathbf{G}$ , following a distribution  $f_{\mathbf{G}}$  similar to the target distribution  $f_{\mathbf{V}}$ , *i.e.* characterized by the parameters  $\mathbf{M}_{\mathbf{V}}$ ,  $m_{\mathbf{V}}$  and  $\beta_{\mathbf{V}}$ . This transformation is denoted by  $T_{\mathbf{U} \rightarrow \mathbf{V}}$ . In the multivariate Gaussian case, *i.e.* when  $\beta_{\mathbf{U}} = \beta_{\mathbf{V}} = 1$ , the Cholesky decomposition, the square root decomposition and the linear Monge-Kantorovitch solution can be used to transform one MGD into another one (see [34] for more details). To the best of our knowledge, when no constraint on the shape parameter  $\beta$  is imposed, a general transformation has not yet been tackled.

We propose a transformation  $T_{\mathbf{U} \rightarrow \mathbf{V}}$ , transforming the input distribution  $f_{\mathbf{U}}$  according to the target distribution  $f_{\mathbf{V}}$ , which consists of two main steps, as illustrated in figure 6.2:

- Monge-Kantorovich closed-form mapping is used to transform the input set  $\mathbf{U}$  into a new set  $\mathbf{W}$ , distributed according to an MGGD of parameters  $\mathbf{M}_{\mathbf{V}}$ ,  $m_{\mathbf{V}}$  and  $\beta_{\mathbf{U}}$  (the input shape parameter  $\beta_{\mathbf{U}}$  remains unchanged). The distribution of the set  $\mathbf{W}$  is denoted by  $f_{\mathbf{W}}$ .
- A stochastic-based transformation of the sample vectors of  $\mathbf{W}$  alters the shape parameter of  $f_{\mathbf{W}}$  with regards to the target shape parameter  $\beta_{\mathbf{V}}$ . Finally, we obtain a new set  $\mathbf{G}$  of sample vectors, following a distribution similar to the target distribution  $f_{\mathbf{V}}$ .

Hereafter, we present details regarding the two steps of our transformation.

### 3.2.1 Monge-Kantorovich closed-form mapping

To transform the input set  $\mathbf{U}$  into a new set distributed according to an MGGD of parameters  $\mathbf{M}_{\mathbf{V}}$ ,  $m_{\mathbf{V}}$  and  $\beta_{\mathbf{U}}$ , we use the Monge-Kantorovich closed-form solution as proposed by Pitié et al. [34, 48]. As a solution of the Monge-Kantorovich mass transportation problem [47], it transfers an input probability density function into another probability density function by minimizing the displacement cost. Given the parameters of the target distribution  $f_{\mathbf{V}}$ , the transformation of the set  $\mathbf{U}$  is defined as:

$$\forall i \in \{1, \dots, N\} : \mathbf{T}_{MK} : \mathbf{u}_i \mapsto \Lambda \mathbf{u}_i, \text{ where} \quad (6.4)$$

$$\Lambda = \left( \frac{m_{\mathbf{V}}}{m_{\mathbf{U}}} \right)^{1/2} \mathbf{M}_{\mathbf{U}}^{-\frac{1}{2}} \left( \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \mathbf{M}_{\mathbf{V}} \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \right)^{\frac{1}{2}} \mathbf{M}_{\mathbf{U}}^{-\frac{1}{2}}. \quad (6.5)$$

Let  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_N)$  be the new set of sample vectors of a random vector  $\mathbf{w}$ , where  $\mathbf{w}_i = \mathbf{T}_{MK}(\mathbf{u}_i)$ ,  $\forall i \in \{1, \dots, N\}$ . The distribution  $f_{\mathbf{W}}$  of the random vector  $\mathbf{w}$  follows the law of an MGGD with a scatter matrix  $\mathbf{M}_{\mathbf{V}}$ , a scale parameter  $m_{\mathbf{V}}$  and a shape parameter  $\beta_{\mathbf{U}}$ . The input shape parameter  $\beta_{\mathbf{U}}$  remains unaffected by the transformation (6.4), as demonstrated in the following section. To transform  $\beta_{\mathbf{U}}$  according to the target shape parameter  $\beta_{\mathbf{V}}$ , we transform the random vectors in the stochastic representation of  $\mathbf{w}$ . This transformation is presented in the following section.

### 3.2.2 Stochastic-based shape parameter transfer

The Monge-Kantorovich transformation (6.4) does not modify the input shape parameter  $\beta_{\mathbf{U}}$  of the distribution  $f_{\mathbf{W}}$ . To demonstrate this fact, we first discuss some important results in connection with the MGGD. As shown in [22, 73, 83, 84], a random vector  $\mathbf{z} \in \mathbb{R}^p$ , distributed according to an MGGD, *i.e.*  $\mathbf{z} \sim MGGD(\mathbf{M}, m, \beta)$ , admits the following stochastic representation:

$$\mathbf{z} \stackrel{d}{=} \tau \cdot (m\mathbf{M})^{\frac{1}{2}} \mathbf{e}, \quad (6.6)$$

where  $\stackrel{d}{=}$  refers to an equality in distribution. The random vector  $\mathbf{e}$  is a unit vector uniformly distributed over a  $p$ -dimensional unit sphere (and thus, independent of  $\beta$ ), whereas  $\tau$  is a scalar positive random variable such that

$$\tau^{2\beta} \sim \Gamma\left(\frac{p}{2\beta}, 2\right), \quad (6.7)$$

where  $\Gamma(a, b)$  denotes the gamma distribution with shape and scale parameters  $a$  and  $b$ , respectively. From equation (6.6),  $\tau$  can be expressed as follows:

$$\tau^2 \stackrel{d}{=} \mathbf{z}^T (m\mathbf{M})^{-1} \mathbf{z}. \quad (6.8)$$

Now, using equations (6.8) and (6.4), we derive the following equality in distribution (where we use the fact that the matrices  $\mathbf{M}_{\mathbf{U}}$  and  $\mathbf{M}_{\mathbf{V}}$  are symmetrical):

$$\begin{aligned} \tau_{\mathbf{w}}^2 &\stackrel{d}{=} \mathbf{w}^T (m_{\mathbf{V}}\mathbf{M}_{\mathbf{V}})^{-1} \mathbf{w} = (\mathbf{T}_{MK}(\mathbf{u}))^T (m_{\mathbf{V}}\mathbf{M}_{\mathbf{V}})^{-1} \mathbf{T}_{MK}(\mathbf{u}) \\ &= \frac{m_{\mathbf{V}}}{m_{\mathbf{U}}} (m_{\mathbf{V}})^{-1} \mathbf{u}^T \mathbf{M}_{\mathbf{U}}^{-1} \left( \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \mathbf{M}_{\mathbf{V}} \mathbf{M}_{\mathbf{V}}^{-1} \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \right) \mathbf{M}_{\mathbf{U}}^{-1} \mathbf{u} = \mathbf{u}^T (m_{\mathbf{U}}\mathbf{M}_{\mathbf{U}})^{-1} \mathbf{u} \stackrel{d}{=} \tau_{\mathbf{u}}^2, \end{aligned} \quad (6.9)$$

where  $\tau_{\mathbf{u}}$  and  $\tau_{\mathbf{w}}$  are the random scalars in the stochastic representations of  $\mathbf{u}$  and  $\mathbf{w}$  respectively. Equation (6.9) indicates that the Monge-Kantorovich transformation (6.4) does not affect  $\tau_{\mathbf{w}}$  and thus, the shape parameter of  $f_{\mathbf{W}}$  remains unchanged after the first step of our transformation. Therefore, to compute a distribution, similar to the target distribution  $f_{\mathbf{V}}$ , we now need to reshape the distribution  $f_{\mathbf{W}}$  of the set  $\mathbf{U}$  using the target shape parameter  $\beta_{\mathbf{V}}$ .

To this end, we transform the set  $\mathbf{W}$  into an output set  $\mathbf{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_i\}$ , where  $\mathbf{g}_i$ ,  $i \in \{1, \dots, N\}$ , are the sample vectors of a random vector  $\mathbf{g}$ , so that the distribution  $f_{\mathbf{G}}$  of  $\mathbf{G}$  is similar to the target distribution  $f_{\mathbf{V}}$ , *i.e.*  $\mathbf{G} \sim MGGD(\mathbf{M}_{\mathbf{V}}, m_{\mathbf{V}}, \beta_{\mathbf{V}})$ . Using equation (6.6), the stochastic representations of the random vectors  $\mathbf{w}$  and  $\mathbf{g}$  are derived as follows:

$$\mathbf{w} \stackrel{d}{=} \tau_{\mathbf{w}} \cdot (m_{\mathbf{V}}\mathbf{M}_{\mathbf{V}})^{\frac{1}{2}} \mathbf{e}_{\mathbf{w}}, \quad (6.10)$$

$$\mathbf{g} \stackrel{d}{=} \tau_{\mathbf{g}} \cdot (m_{\mathbf{V}}\mathbf{M}_{\mathbf{V}})^{\frac{1}{2}} \mathbf{e}_{\mathbf{g}}. \quad (6.11)$$

As the sample vectors  $\mathbf{w}_i$  are known, the equality in distribution (6.10) can be transformed into a strict equality in values for the following pair of sample vectors, denoted by  $\hat{\tau}_{\mathbf{w}_i}$  and  $\hat{\mathbf{e}}_{\mathbf{w}_i}$  (a demonstration is provided in appendix A):

$$\hat{\tau}_{\mathbf{w}_i} = \left\| (m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{w}_i \right\|, \quad (6.12)$$

$$\hat{\mathbf{e}}_{\mathbf{w}_i} = \frac{(m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{w}_i}{\left\| (m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{w}_i \right\|}. \quad (6.13)$$

The vectors  $\tau_{\mathbf{g}}$  and  $\mathbf{e}_{\mathbf{g}}$  cannot be obtained in a similar manner, because the sample vectors  $\mathbf{g}_i$  are unknown. However, to generate the random vector  $\mathbf{g}$  using its stochastic representation, we first need to compute the random vectors  $\tau_{\mathbf{g}}$  and  $\mathbf{e}_{\mathbf{g}}$ .

The key idea of the proposed transformation consists in transforming  $\hat{\tau}_{\mathbf{w}_i}$  and  $\hat{\mathbf{e}}_{\mathbf{w}_i}$  into a sample scalar  $\hat{\tau}_{\mathbf{g}_i}$  and a sample unit vector  $\hat{\mathbf{e}}_{\mathbf{g}_i}$  respectively. The transformation of  $\hat{\tau}_{\mathbf{w}}$  and  $\hat{\mathbf{e}}_{\mathbf{w}}$  aims to modify the shape parameter  $\beta_{\mathbf{U}}$  of distribution  $\mathbf{f}_{\mathbf{W}}$  with respect to the target shape parameter  $\beta_{\mathbf{V}}$ . As only  $\hat{\tau}_{\mathbf{w}_i}$  depends on  $\beta_{\mathbf{U}}$ , we transform  $\hat{\tau}_{\mathbf{w}_i}$  and keep  $\hat{\mathbf{e}}_{\mathbf{w}_i}$  unchanged (*i.e.*  $\hat{\mathbf{e}}_{\mathbf{g}_i} = \hat{\mathbf{e}}_{\mathbf{w}_i} \forall i \in \{1, \dots, N\}$ ). We present the transformation of the sample vector  $\hat{\tau}_{\mathbf{w}_i}$  in the following proposition (the proof is given in the appendix B).

**Proposition 3.2.1.** *Let  $\hat{\tau}_{\mathbf{w}}$  and  $\hat{\tau}_{\mathbf{g}}$  denote scalar positive random variables, whose distributions are defined as follows,  $\beta_{\mathbf{U}} \in \mathbb{R}$ ,  $\beta_{\mathbf{V}} \in \mathbb{R}$  and  $p \in \mathbb{N}$ :*

$$\hat{\tau}_{\mathbf{w}}^{2\beta_{\mathbf{U}}} \sim \Gamma\left(\frac{p}{2\beta_{\mathbf{U}}}, 2\right), \quad (6.14)$$

$$\hat{\tau}_{\mathbf{g}}^{2\beta_{\mathbf{V}}} \sim \Gamma\left(\frac{p}{2\beta_{\mathbf{V}}}, 2\right). \quad (6.15)$$

Then, the sample vector  $\hat{\tau}_{\mathbf{g}_i}$  of the variable  $\hat{\tau}_{\mathbf{g}}$  can be obtained by transforming the sample vector  $\hat{\tau}_{\mathbf{w}_i}$  using the following approximation:

$$\hat{\tau}_{\mathbf{g}_i} = \left( \frac{(\hat{\tau}_{\mathbf{w}_i})^{2\lambda\beta_{\mathbf{U}}} - \mu_{\lambda}}{\sigma_{\lambda}} \sigma'_{\lambda} + \mu'_{\lambda} \right)^{\frac{\lambda}{2\beta_{\mathbf{V}}}}, \quad (6.16)$$

where the scalar variables  $\mu_{\lambda}$  and  $\sigma_{\lambda}$  (respectively  $\mu'_{\lambda}$  and  $\sigma'_{\lambda}$ ) are the mean and the standard deviation of a univariate normal distribution with the following parameters [85]:

$$\mu_{\lambda} = 2^{\lambda} \frac{\Gamma(\beta + \lambda)}{\Gamma(\beta)} \text{ and } \sigma_{\lambda}^2 = 2^{2\lambda} \frac{\Gamma(\beta + 2\lambda)}{\Gamma(\beta)} - \mu_{\lambda}^2 \quad (6.17)$$

for  $\beta = p/(2\beta_{\mathbf{U}})$  (respectively  $\beta' = p/(2\beta_{\mathbf{V}})$  for  $\mu'_{\lambda}$  and  $\sigma'_{\lambda}$ ) and for  $\lambda = 1/4$ . The approximation (6.16) is valid under the following two assumptions:

$$\Gamma\left(\frac{p}{2\beta_{\mathbf{U}}}, 2\right) \sim \Gamma\left(\left[\frac{p}{2\beta_{\mathbf{U}}}\right], 2\right), \quad (6.18)$$

$$\Gamma\left(\frac{p}{2\beta_{\mathbf{V}}}, 2\right) \sim \Gamma\left(\left[\frac{p}{2\beta_{\mathbf{V}}}\right], 2\right), \quad (6.19)$$

where  $[\cdot]$  denotes the nearest integer value (see appendix B for further details and proof). ■

The transformation (6.16) holds for each sample value of the scalar variable  $\hat{\tau}_{\mathbf{w}}$  with a distribution, given by (6.14). However, in our method, the vector  $\hat{\tau}_{\mathbf{w}}$  is constructed uniquely using equation (6.12).

Furthermore, we can now define the stochastic-based transformation of the distribution  $f_{\mathbf{w}}$ :

$$\forall i \in \{1, \dots, N\}, \mathbf{T}_S : \mathbf{w}_i \mapsto h(\mathbf{w}_i), \text{ where} \quad (6.20)$$

$$h(\mathbf{w}_i) = \hat{\tau}_{\mathbf{g}_i} (m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{\frac{1}{2}} \hat{\mathbf{e}}_{\mathbf{w}_i}. \quad (6.21)$$

The sample scalar  $\hat{\tau}_{\mathbf{g}_i}$  is computed with equation (6.16) and the unit vector  $\hat{\mathbf{e}}_{\mathbf{w}_i}$  is computed with equation (6.13). By applying transformation (6.20), we obtain the new set  $\mathbf{G}$  of sample vectors, given as follows:  $\mathbf{g}_i = \mathbf{T}_S(\mathbf{w}_i)$ ,  $\forall i \in \{1, \dots, N\}$ . The set  $\mathbf{G}$  is approximately distributed as the target MGGD  $f_{\mathbf{V}}$ .

### 3.2.3 The final transformation $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$

In the present section we introduce the final transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  between the distributions of the sets  $\mathbf{U}$  and  $\mathbf{V}$ . The transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is obtained as a combination of the transformations  $\mathbf{T}_{MK}$  and  $\mathbf{T}_S$  as follows:

$$\forall i \in \{1, \dots, N\}, \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}} : \mathbf{u}_i \mapsto \mathbf{T}_S [\mathbf{T}_{MK}(\mathbf{u}_i)]. \quad (6.22)$$

Transformation (6.22) is equivalent to:

$$\forall i \in \{1, \dots, N\}, \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}} : \mathbf{u}_i \mapsto \mathbf{K}_i \mathbf{u}_i, \text{ where} \quad (6.23)$$

$$\mathbf{K}_i = \frac{(m_{\mathbf{V}}/m_{\mathbf{U}})^{\frac{1}{2}} \hat{\tau}_{\mathbf{g}_i}}{\|(m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{w}_i\|} \mathbf{M}_{\mathbf{U}}^{-\frac{1}{2}} \left( \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \mathbf{M}_{\mathbf{V}} \mathbf{M}_{\mathbf{U}}^{\frac{1}{2}} \right)^{\frac{1}{2}} \mathbf{M}_{\mathbf{U}}^{-\frac{1}{2}}. \quad (6.24)$$

The sample scalar  $\hat{\tau}_{\mathbf{g}_i}$  is computed with transformation (6.16), in which the vector  $\hat{\tau}_{\mathbf{w}_i}$  is replaced by expression (6.12). The proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is used to generate the new set of sample vectors  $\mathbf{G} = (\mathbf{g}_1, \dots, \mathbf{g}_N)$ ,  $\mathbf{g}_i = \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\mathbf{u}_i)$ ,  $\forall i \in \{1, \dots, N\}$ , the distribution of which follows the law of an MGGD and is similar to the distribution of the target set  $\mathbf{V}$ .

Algorithm 2 illustrates the main steps of our transformation through a pseudo code, which facilitates the implementation of the method.

### 3.2.4 Properties of the transformation $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$

The properties of the transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  are listed below:

- P1:** The matrix  $\mathbf{K}_i \in \mathbb{R}^{p \times p}$ ,  $\forall i \in \{1, \dots, N\}$ ,  $p \in \mathbb{N}$ , is a positive definite matrix, i.e.  $\mathbf{x}^T \mathbf{K}_i \mathbf{x} > 0 \forall \mathbf{x} \neq \mathbf{0}$ ;
- P2:** The transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is invertible as  $\forall i \in \{1, \dots, N\}$  the matrix  $\mathbf{K}_i$  is invertible (as a consequence of property **P1**). The inverse transform  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}^{-1}$  is given as:  $\forall i \in \{1, \dots, N\}$ ,  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}^{-1} : \mathbf{g}_i \mapsto \mathbf{K}_i^{-1} \mathbf{g}_i$ , e.g.  $\mathbf{u}_i = \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}^{-1}(\mathbf{g}_i)$ ;
- P3:** The identity vector  $\mathbf{1}$  is the identity element of the transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$ ;
- P4:** When  $\beta_{\mathbf{U}} = \beta_{\mathbf{V}}$ , the transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is equivalent to the Monge-Kantorovich closed-form transformation  $\mathbf{T}_{MK}$ , defined in (6.4);



**Algorithm 2** The transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$ 


---

```

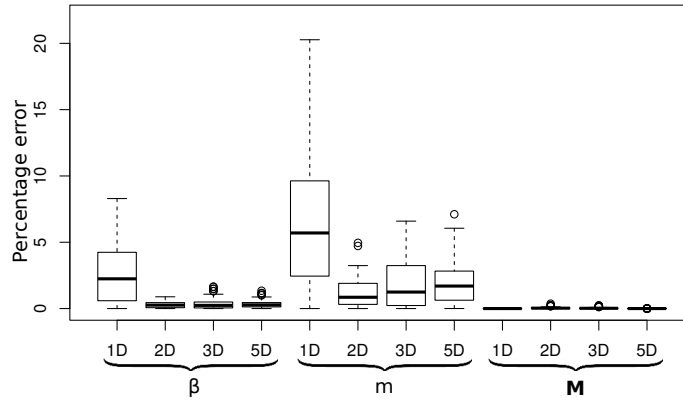
1: procedure TRANSFORMMGGD
2: input:
3:    $\mathbf{U} \leftarrow (\mathbf{u}_1, \dots, \mathbf{u}_N)$  ▷ input
4:    $\mathbf{V} \leftarrow (\mathbf{v}_1, \dots, \mathbf{v}_L)$  ▷ target
5:    $p \leftarrow \text{GetDimension}(\mathbf{U})$ 
6:    $i \leftarrow 0$  ▷ loop index
7:    $\lambda \leftarrow 1/4$  ▷ used in eqn. 6.16
8: output:
9:    $\mathbf{G} \leftarrow (\mathbf{g}_1, \dots, \mathbf{g}_N)$  ▷ computed in loop
10: parameters: ▷ eqn. 6.1, 6.2 and 6.3
11:    $(\mathbf{M}_{\mathbf{U}}, m_{\mathbf{U}}, \beta_{\mathbf{U}}) = \text{ComputeMLE}(\mathbf{U})$ 
12:    $(\mathbf{M}_{\mathbf{V}}, m_{\mathbf{V}}, \beta_{\mathbf{V}}) = \text{ComputeMLE}(\mathbf{V})$ 
13: loop:
14:    $\mathbf{w}_i \leftarrow \text{MongeKantorovich}(\mathbf{u}_i, \mathbf{M}_{\mathbf{U}}, \mathbf{M}_{\mathbf{V}}, m_{\mathbf{U}}, m_{\mathbf{V}})$  ▷ transf. 6.4
15:    $(\hat{\tau}_{\mathbf{w}_i}, \hat{\epsilon}_{\mathbf{w}_i}) \leftarrow \text{ConstructStochasticVectors}(\mathbf{w}_i, \mathbf{M}_{\mathbf{V}}, m_{\mathbf{V}})$  ▷ eqn. 6.12 and 6.13
16:    $\hat{\tau}_{\mathbf{g}_i} \leftarrow \text{TransformTau}(\hat{\tau}_{\mathbf{w}_i}, \beta_{\mathbf{U}}, \beta_{\mathbf{V}}, p, \lambda)$  ▷ eqn. 6.16
17:    $\mathbf{g}_i \leftarrow \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\mathbf{w}_i, \hat{\tau}_{\mathbf{g}_i}, \mathbf{M}_{\mathbf{U}}, \mathbf{M}_{\mathbf{V}}, m_{\mathbf{U}}, m_{\mathbf{V}})$  ▷ transf. 6.23
18:    $i \leftarrow i + 1$ 
19:   if  $i < N$  then
20:     goto loop

```

---

**P5:** When the dimension  $p = 1$ ,  $\mathbf{K}_i = [c_i]$ , where  $c_i$  is a scalar  $\forall i \in \{1, \dots, N\}$ ;

**P6:** When the dimension  $p = 1$  and  $\beta_{\mathbf{U}} = \beta_{\mathbf{V}}$ ,  $\mathbf{K}_i = [c] \forall i \in \{1, \dots, N\}$ , where  $c = (m_{\mathbf{V}}/m_{\mathbf{U}})^{\frac{1}{2}}$ . Therefore, in this case, the proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is equivalent to scaling the input set  $\mathbf{U}$  by the constant positive scalar  $c$ .



**Figure 6.3** – Box-and-Whisker plots of the percentage error of the proposed transformation, shown parameter-wise for 1D, 2D, 3D and 5D spaces. The mean percentage error in the 2D, 3D and 5D cases is less than 5% for the scale parameter  $m$  and less than 2% for both the shape parameter  $\beta$  and the scatter matrix  $\mathbf{M}$ .

### 3.2.5 Evaluation

We demonstrate the efficiency of our method in 2D ( $p = 2$ ), 3D ( $p = 3$ ) and 5D ( $p = 5$ ) as these are the dimensions directly connected to the applications, presented in this chapter (see section 4). For completeness sake, we also show the performance of the proposed method in 1D ( $p = 1$ ). To evaluate the performance of the proposed transformation, we



generate 10 MGGDs per dimension  $p$ . We randomly generate all MGGDs used in our evaluation by applying the stochastic representation of the MGGD (equation (6.6)). Once we initialize the parameters  $\mathbf{M}$ ,  $m$  and  $\beta$  of an MGGD by  $\mathbf{M}_0$ ,  $m_0$  and  $\beta_0$ , we randomly generate a  $p$ -dimensional unit vector  $\mathbf{e}$ , uniformly distributed on a unit sphere ( $p$  stands for either 1, 2, 3 or 5), and its corresponding scalar random variable  $\tau$ . By applying the stochastic representation (6.6), we compute a  $p$ -dimensional vector  $\mathbf{z}$ , distributed according to an MGGD of parameters  $\mathbf{M}_0$ ,  $m_0$  and  $\beta_0$ .

We have carried out the proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  between each two of the generated distributions for each dimension, resulting in a total of 100 transformations per case. To illustrate the robustness of the transformation, percentage errors, *i.e.* the ratio of the difference between the target and the transformed parameters to the target parameter, have been computed. Figure 6.3 presents the Box-and-Whisker plots of these percentage errors parameter-wise for each of the four cases. The proposed transformation manages to transform the shape parameter  $\beta$  and the scatter matrix  $\mathbf{M}$  with a great precision for the 2D, 3D and 5D cases: both the mean percentage error and the standard deviation are less than 2%. The percentage error of the scatter matrix  $\mathbf{M}$  in the 1D case is constantly 0, as both input and target matrices are always equal to the 1D identity matrices. Finally, the transformation between two 1D MGGDs is far less accurate than the transformation between any two  $p$ -dimensional MGGDs  $\forall p > 1$ . This is due to the fact that for  $p = 1$ ,  $p/(2\beta) = 1/(2\beta) \geq 0.5$ , *i.e.* the shape parameter of the gamma distributions of  $\hat{\tau}_w$  and  $\hat{\tau}_g$  could become smaller than 1. This makes assumptions (6.18) and (6.19) in proposition 3.2.1 too strong for  $p = 1$  (see appendix B for demonstration).

## 4 Applications

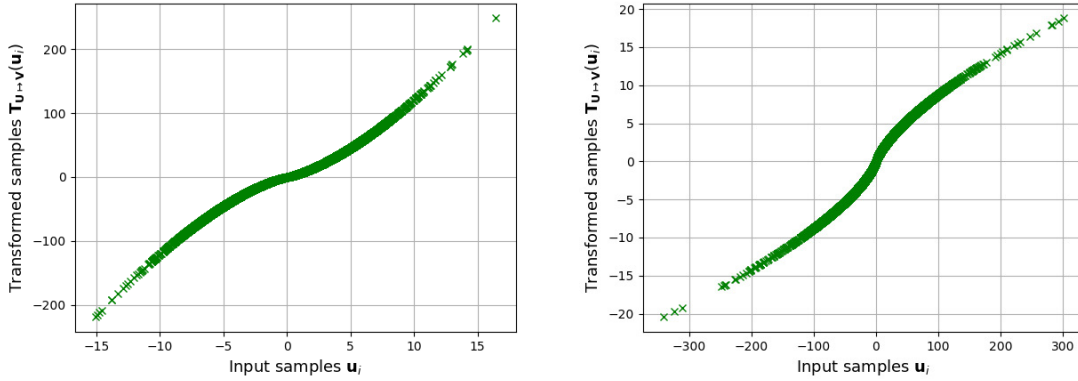
In this section, we demonstrate the potential of the proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  for several image processing applications. First, we carry out a transfer of gradient and a transfer of color between images. Second, as our method describes the joint distribution of color and gradient by one probabilistic model, we carry out a simultaneous transfer of color and gradient. We apply our simultaneous transfer for image color correction.

Before presenting the applications of our transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$ , we first describe what properties  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  should have in order to provide plausible results for color and gradient transfers.

**Continuity.** When transferring color and gradient between images, the geometry of the input distribution needs to be respected. To this end, similar colors in the input image should remain similar after the transformation. We call this property continuity.

The first step of the transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$ , *i.e.* the Monge-Kantorovich transformation, ensures continuity as it keeps the geometry of the input image as intended [34]. The stochastic-based transformation also preserves the continuity of our transformation. To demonstrate this fact, we plot samples  $\mathbf{u}_i$  against transformed samples  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\mathbf{u}_i)$  as shown in figure 6.4. From the two plots in figure 6.4 it becomes clear that our transformation is continuous. Moreover, the plots show that our transformation is non-linear.

**Bounded range.** As the image color distributions are bounded in a discrete interval, the transformed color distributions should also be bounded. The violation of this property would cause inconsistencies in the results, such as color artifacts, out-of-gamut pixels, etc.



**Figure 6.4** – We carry out our 1D MGGD-based transformation for two different sets of input samples and we plot the input samples against the transformed samples. These plots show that our transformation is both continuous and non-linear.

Our transformation  $T_{U \rightarrow V}$  maps the input range to the target range (this can be observed in the plots of figures 6.2 and 6.5). Therefore, if the target range is bounded, the value range of our results is also bounded.

The described properties are essential for obtaining plausible results from applying our transformation  $T_{U \rightarrow V}$  to image processing applications. Four applications are presented in the following sections.

## 4.1 Gradient transfer

As we adopt our transformation to perform a gradient transfer between images, we first present previous works, related to modifying the gradient field of an image. The related works consist of methods which directly alter an input gradient field with regards to a target one as well as of methods which perform a transfer between gradient distributions. We then introduce our MGGD-based method for carrying out a gradient transfer between images.

**Poisson equation and gradient boosting.** The Poisson equation is adopted in various image processing applications such as seamless cloning [86, 87], image sharpening [88], image stylization [89], Poisson image editing [86], tone mapping of HDR images [90]. The Poisson equation [86] introduces a solution for constructing an image given a gradient field. The construction is carried out by minimizing the distance between the gradient field of the image that we recover and the given gradient field. The latter is either known or it is computed by scaling the image gradient field by a constant. The aforementioned method is referred to as gradient boosting [88, 91] and it aims at amplifying the gradient field of an image by a constant value. Gradient boosting can be applied globally to the whole image as well as locally for the purpose of sharpening the salient parts of an image [91].

Furthermore, a modification of the Poisson equation, in which a new data term is added, is well-known as the screened Poisson equation [88, 92]. The new data term constrains the Poisson equation, allowing the reconstructed image to be as close as possible to a given image. Bhat *et al.* [88] have introduced a Fourier solution to the screened Poisson equation, adopted later on by Morel *et al.* [92] in connection with the image contrast enhancement.

**Gradient transfer using a target gradient field.** Dend *et al.* [91] sharpen an input image by using the gradient field of its corresponding near-to-infrared (NIR) image. As the NIR images contain more details than the RGB images, their gradients are much shaper. Therefore, Deng *et al.*'s color-aware regularization manages to recover the details of the input image. Furthermore, Cho *et al.* [25] reconstruct a high-quality image from a degraded image by matching gradient distributions. Univariate generalized Gaussian distributions (GGDs) are adopted to model the distributions of each of the two components of the input and target gradient fields. The target gradient is computed by de-convolving and down-sampling the input image. Cho *et al.* recover a high-quality result by minimizing the Kullback-Leibler divergence between the input and target gradient distributions either by penalizing the divergence or by reweighing the distributions. Unlike Cho *et al.* [25], who model the distribution of the gradient field of images, Gong *et al.* [36] introduce a new parametric model for approximating the gradient cumulative density function. By empirically learning the target gradient distribution from a database of natural images, the authors aim at naturalizing a given input image.

So far, a transfer of gradient has only been carried out for target distributions which are computed either by down-sampling the input image [25] or by using a database of natural images [36]. However, a gradient transfer, imposing no restrictions on the choice of the target images, has not yet been introduced. To address the latter limitation, we propose an MGGD-based transfer of gradient between images, as discussed hereafter.

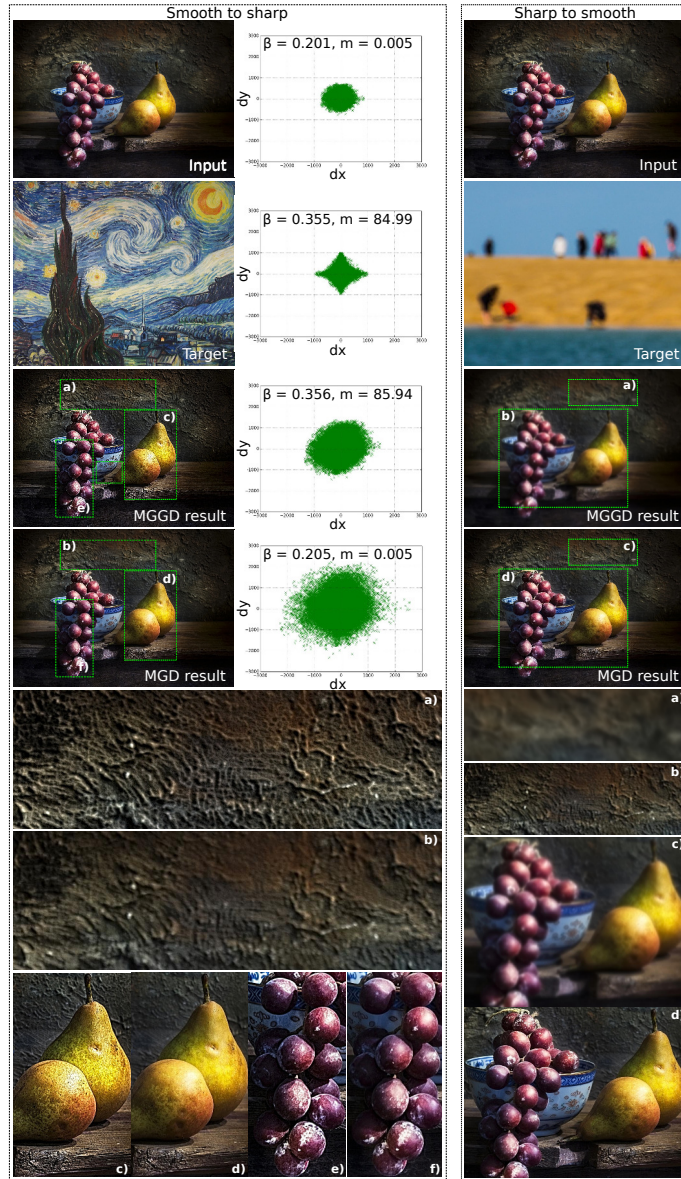
**MGGD-based gradient transfer (2D).** We employ our transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  in the context of a gradient transfer between an input image  $\mathbf{U} : \omega \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$  and a target image  $\mathbf{V} : \omega \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$  with gradient fields  $\nabla \mathbf{U}$  and  $\nabla \mathbf{V}$  respectively. The parameters of both gradient distributions  $f_{\nabla \mathbf{U}}$  and  $f_{\nabla \mathbf{V}}$  are computed using equations (6.1), (6.2) and (6.3). Our transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  transforms the input gradient distribution  $f_{\nabla \mathbf{U}}$  into a distribution similar to the target gradient distribution  $f_{\nabla \mathbf{V}}$ .

We perform the gradient transfer to either sharpen or smoothen the input gradient field  $\nabla \mathbf{U}$  with regards to the target gradient field  $\nabla \mathbf{V}$ . Unlike Cho *et al.* [25], we model the input and target gradient fields by 2D MGGDs. On one hand, we take into account the correlation between the two components of the gradient field for the input and target images since we transform the joint  $\mathbf{xy}$  gradient distribution rather than the marginal distributions of  $\mathbf{x}$  and  $\mathbf{y}$ . On the other hand, the proposed transformation is more accurate for 2D MGGDs than for 1D MGGDs as illustrated in figure 6.3. In practice, the gradient transfer is carried out on the luminance channel of CIE Lab color space (unless mentioned otherwise).

We aim to recover an image  $\mathbf{F} : \omega \subset \mathbb{R}^2 \mapsto \mathbb{R}^3$  (where  $\omega = \omega_x \times \omega_y$ ), similar to the input image  $\mathbf{U}$ , whose gradient field is as close as possible to the transformed gradient field  $\mathbf{G} = \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\nabla \mathbf{U})$ . The image  $\mathbf{F}$  is obtained by minimizing the following double integral:

$$\min \int_{\omega_x} \int_{\omega_y} \lambda_s \|\mathbf{F} - \mathbf{U}\|^2 + \|\nabla \mathbf{F} - \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\nabla \mathbf{U})\|^2 d\mathbf{x} d\mathbf{y} \quad (6.25)$$

where  $\nabla \mathbf{F}$  is the gradient field of  $\mathbf{F}$ . The constant  $\lambda_s \in \mathbb{R}$  controls the trade-off between the data term and the transformed gradient field. The solution of the minimization problem (6.25) is given by the screened Poisson equation [88].



**Figure 6.5** – Results from a gradient transfer between images for two cases: image sharpening (smooth to sharp) and image smoothing (sharp to smooth). In the case of smooth to sharp transfer, our transformation enhances the input details, sharpens the foreground and the background of the input image. The MGD-based transformation does not influence significantly the sharpness of the input gradient field, yielding a result, similar to the input image (the sharpness of the background remains relatively unchanged). In the case of sharp to smooth transfer, our gradient transfer simulates the blurry effect of the target image. In contrast the MGD-based transfer preserves the input details and sharpness. Quantitative differences between the two results are illustrated by the distribution plots (for the smooth to sharp transfer), where  $dx$  and  $dy$  denote the gradient in  $x$  and  $y$  directions respectively. The MGD-based transformation spreads out the input distribution (note the range of the axes), as it does not affect the shape parameter  $\beta$ . In contrast, our method transforms both the scale and the shape of the input gradient distribution. As the target shape is very specific (*i.e.* diamond shape), we do not transfer the shape precisely (as illustrated by the plots), but we obtain a distribution, which can well be characterized by the target shape and scale parameters (note also the ranges of the distributions of our result and the target image).



The new gradient field  $\mathbf{G}$  is obtained pixel-wise by multiplying the input gradient field  $\nabla \mathbf{U}$  by the transformation matrix  $\mathbf{K}_i$ ,  $\forall i \in \{1, \dots, N\}$ , where  $N$  is the number of pixels in  $\mathbf{F}$  (as presented in (6.24)). To this end, the advantage of our method over the method of gradient boosting becomes clear. Rather than scaling the input gradient field  $\nabla \mathbf{U}$  by a given constant  $c$ , we automatically transform each pixel of  $\mathbf{U}$  with respect to the characteristics of the target gradient distribution. The pseudo code implementation of our gradient transfer is presented in algorithm 3.

---

**Algorithm 3** Gradient transfer (2D)

---

```

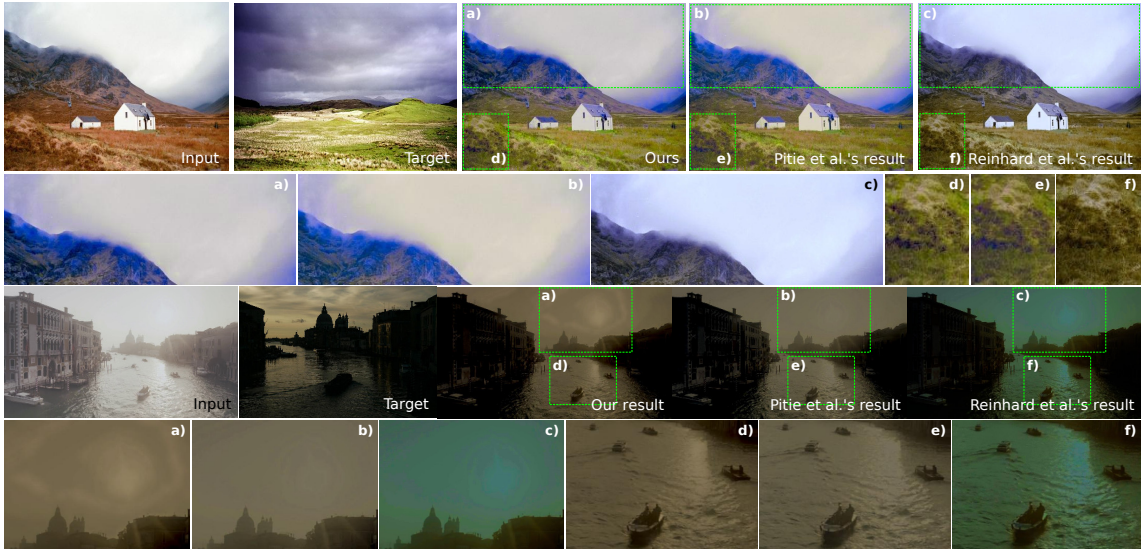
1: procedure PERFORMGRADIENTTRANSFER
2: input:
3:    $\mathbf{U} \leftarrow (\mathbf{u}_1, \dots, \mathbf{u}_N)$  ▷ input image
4:    $\mathbf{V} \leftarrow (\mathbf{v}_1, \dots, \mathbf{v}_L)$  ▷ target image
5:    $p \leftarrow 2$  ▷ for 2D gradient fields
6:    $\lambda_s \leftarrow 0.2$  ▷ used in eqn. 6.25
7: gradients:
8:    $\nabla \mathbf{U} \leftarrow \text{ComputeGradientField}(\mathbf{U})$ 
9:    $\nabla \mathbf{V} \leftarrow \text{ComputeGradientField}(\mathbf{V})$ 
10: transformed gradient:
11:    $\mathbf{G} \leftarrow (\mathbf{g}_1, \dots, \mathbf{g}_N)$  ▷ computed in loop
12: output:
13:    $\mathbf{F} \leftarrow (\mathbf{f}_1, \dots, \mathbf{f}_N)$  ▷ computed in construction
14: parameters: ▷ eqn. 6.1, 6.2 and 6.3
15:    $(\mathbf{M}_\mathbf{U}, m_\mathbf{U}, \beta_\mathbf{U}) = \text{ComputeMLE}(\nabla \mathbf{U})$ 
16:    $(\mathbf{M}_\mathbf{V}, m_\mathbf{V}, \beta_\mathbf{V}) = \text{ComputeMLE}(\nabla \mathbf{V})$ 
17: loop: ▷ The transformed gradient field  $\mathbf{G}$  is obtained using the steps of the loop of the algorithm 2
   (lines 14-20) by replacing  $\mathbf{u}_i$  with  $\nabla \mathbf{u}_i$ .
18: construction of the final image  $\mathbf{F}$ :
19:    $\mathbf{F} = \text{ScreenedPoisson}(\mathbf{U}, \mathbf{G}, \lambda_s)$  ▷ eqn. 6.25

```

---

Figure 6.5 shows a comparison between the proposed MGGD-based transfer of gradient and the Monge-Kantorovich transformation (for two types of gradient transfer: smooth to sharp and sharp to smooth). As the latter assumes that the gradient distribution of both input and target images can be fitted by an MGD, its impact on the gradient field of the results is insignificant. Unlike the Monge-Kantorovich transformation, our MGGD-based gradient transfer sharpens the fine details of the input image and enhances its contrast (smooths the details and flattens the input image in the case of sharp to smooth transfer). Moreover, it manages to better transfer the quantitative characteristics of the target distribution, *e.g.* the shape and scale parameters. The shape and scale parameters of the distribution of our result are close to the target ones, as illustrated by the distribution plots in figure 6.5. We obtain a distribution, which can be well described by the target distribution parameters. In contrast, the Monge-Kantorovich transformation spreads out the gradient values of the result (by scaling the input distribution, without modifying its shape parameter), yielding a less sparse distribution than the distribution, obtained with our MGGD-based transformation.

Our gradient transfer can be carried out (independently or jointly) on all three channels of CIE Lab as well as on the channels of any other color space. The scalar  $\lambda_s$  in (6.25) is set to 0.2 for all the results shown in this chapter. Finally, we regularize  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  using a bilateral filter, applied on the set of transformation matrices  $(\mathbf{K}_1, \dots, \mathbf{K}_N)$ . This regularization has been adopted to account for jpeg block artifacts as well as artifacts, occurring



**Figure 6.6** – Results from a color transfer between images, obtained with our 3D MGGD-based method and Pitié’s and Reinhard’s MGD-based methods. The first row presents input and target images, whose color distributions can be well described by an MGD. We observe a slight green cast on the sky in Pitié’s result, which does not appear in our result (as shown in a) and b), second row). Moreover, there is a blue artifact on the grass in Pitié’s result (as shown in c) and d), second row). Reinhard’s result transfers the best the color of the sky, but it fails to transfer the color of the grass. The third row presents an input over-exposed image and a target under-exposed image. The color distributions of these images cannot be accurately fitted by an MGD. We obtain a result, in which the main color of the target light is well transferred. Our method preserves the original details of the sky (as shown in a) and b), forth row). Although the light in Pitié’s result has decreased (in comparison to the input image), the result still looks like an over-exposed image due to the flat areas in the sky and the water. Finally, Reinhard’s result fails to transfer properly the target color palette. The results and the main visual differences between them are best viewed on screen.

with the increase of small gradients. This regularization does not affect significantly the gradient distribution of the result.

## 4.2 Color transfer (3D)

The proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  is also used to carry out a color transfer between an input image  $\mathbf{U}$  and a target image  $\mathbf{V}$ . We model the input/target color distributions by 3D MGGDs, which are computed directly from the color distributions of the input and target images without imposing restrictions on the shape parameter  $\beta$ . That way, we describe more accurately the color distributions of the input and target images and we enhance the quality of the color transfer for distributions which cannot be well-fitted by an MGD. The transformed image  $\mathbf{G}$  is obtained as follows:

$$\mathbf{G} = \mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}} (\mathbf{U} - \boldsymbol{\mu}_{\mathbf{U}}) + \boldsymbol{\mu}_{\mathbf{V}}, \quad (6.26)$$

where  $\boldsymbol{\mu}_{\mathbf{U}}$  and  $\boldsymbol{\mu}_{\mathbf{V}}$  are the sample mean vectors of the input and target distributions respectively. We apply the transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$  on the centered input distribution as the proposed method is built for MGGDs with a zero mean. Comparisons between our 3D

MGGD-based method and the Monge-Kantorovich transformation (carried out in the context of color transfer, as proposed by Pitié *et al.* [34]) are shown in figure 6.6. When the input and target color distributions can be described by an MGGDs with a shape parameter close to 1 (as it is the case of the first pair of input/target images in figure 6.6), our 3D color transfer method yields a visually similar result to the result obtained with Pitié *et al.*'s method (the first two rows in figure 6.6). However, for image color distributions, described by an MGGD with a shape parameter  $\beta \ll 1$ , we observe a significant difference in the color palettes of the two results (the last two rows in figure 6.6). Unlike Pitié *et al.*'s result, in this case our result manages to represent better the main color of the target image, as further discussed in the caption of figure 6.6.

### 4.3 Simultaneous color and gradient transfer (5D)

So far, we have demonstrated the efficiency of the proposed MGGD-based transformation in the context of color and gradient transfers between images. However, both color and gradient have an impact on the style of images. To take the dependencies between color and gradient into account, we model their joint distribution by an MGGD (for both the input and target images). Then, we simultaneously transform the input color and gradient distributions by applying the proposed transformation  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}$ .

The simultaneous transfer between the input and target images is carried out in a 5D space consisting of the three components of CIE Lab color space and the two components of the image gradient field ( $d\mathbf{x}$  and  $d\mathbf{y}$ ). The gradient field of both the input and target images is computed from the L channel of CIE Lab. After performing our transformation, we obtain a new set  $\mathbf{G}$ , consisting of 5D sample vectors. The union of the L, a, b components of each vector in the set  $\mathbf{G}$  corresponds to an image, denoted by  $\mathbf{G}_{lab}$ . The union of the remaining two vector components in  $\mathbf{G}$  corresponds to a gradient field, denoted by  $\mathbf{G}_{xy}$ . To recover the final image  $\mathbf{F}$ , we apply the minimization process in (6.25), where the input image  $\mathbf{U}$  is replaced by the image  $\mathbf{G}_{lab}$  and the expression  $\mathbf{T}_{\mathbf{U} \rightarrow \mathbf{V}}(\nabla \mathbf{U})$  is replaced by the gradient field  $\mathbf{G}_{xy}$  (see algorithm 4 for a pseudo code).

---

#### Algorithm 4 Simultaneous color and gradient transfer (5D)

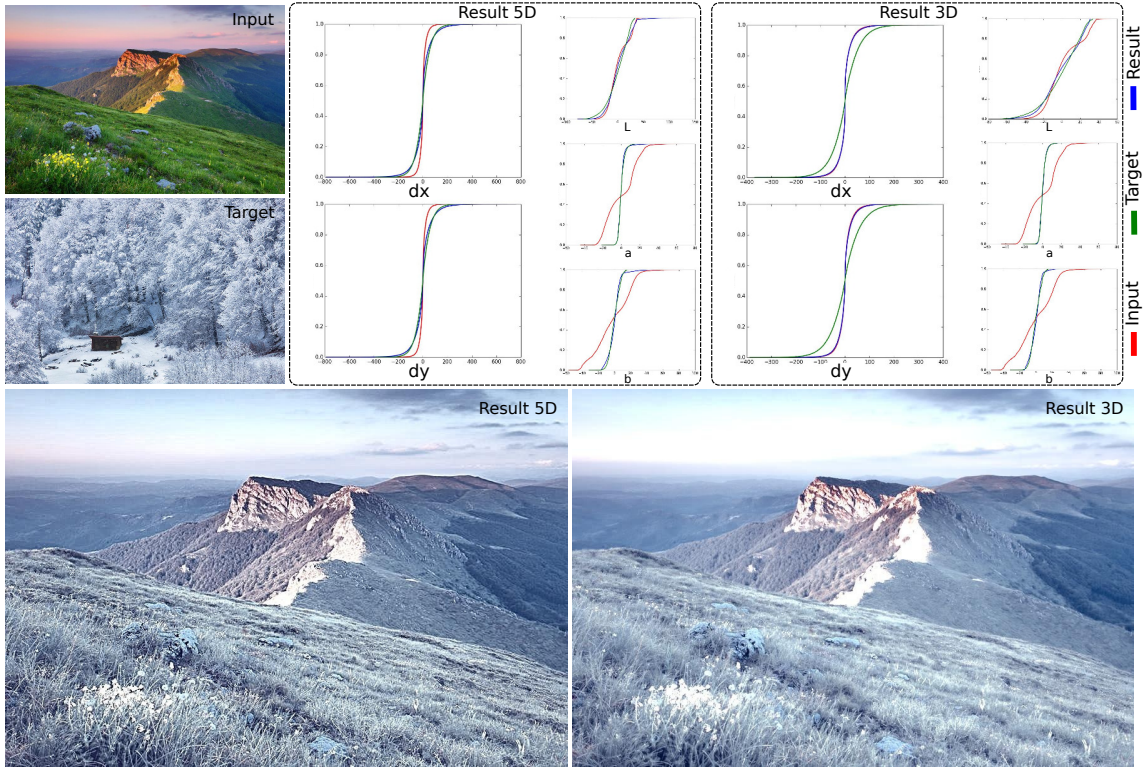
---

```

1: procedure PERFORMSIMULTANEOUSTRANSFER
2:   input:
3:      $\mathbf{U} \leftarrow (\mathbf{u}_1, \dots, \mathbf{u}_N)$  ▷ input image + input gradient
4:      $\mathbf{V} \leftarrow (\mathbf{v}_1, \dots, \mathbf{v}_L)$  ▷ target image + target gradient
5:      $p \leftarrow 5$  ▷ 3D color + 2D gradient field
6:      $\lambda_s \leftarrow 0.2$  ▷ used in eqn. 6.25
7:   transformed gradient:
8:      $\mathbf{G} \leftarrow (\mathbf{g}_1, \dots, \mathbf{g}_N)$  ▷ 5D set, computed in loop
9:   output:
10:     $\mathbf{F} \leftarrow (\mathbf{f}_1, \dots, \mathbf{f}_N)$  ▷ final image, computed in construction
11:   parameters: ▷ eqn. 6.1, 6.2 and 6.3
12:     $(\mathbf{M}_{\mathbf{U}}, m_{\mathbf{U}}, \beta_{\mathbf{U}}) = \text{ComputeMLE}(\mathbf{U})$ 
13:     $(\mathbf{M}_{\mathbf{V}}, m_{\mathbf{V}}, \beta_{\mathbf{V}}) = \text{ComputeMLE}(\mathbf{V})$ 
14:   loop: ▷ The 5D set  $\mathbf{G} = \mathbf{G}_{lab} \cup \mathbf{G}_{xy}$  is obtained using the steps in the loop of the algorithm 2 (lines 14-20).
15:   construction of the final image  $\mathbf{F}$ :
16:     $\mathbf{F} = \text{ScreenedPoisson}(\mathbf{G}_{lab}, \mathbf{G}_{xy}, \lambda_s)$  ▷ eqn. 6.25

```

---

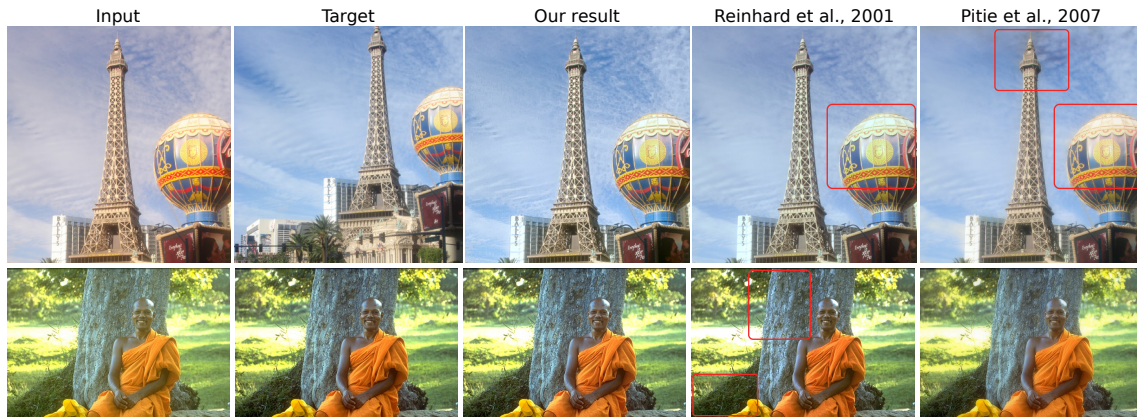


**Figure 6.7** – Result of a simultaneous 5D transfer of color and gradient, compared to a result of a 3D color transfer (obtained with our transformation). As the target image is a sharp detailed image, the 5D transfer amplifies the fine details of the input image (unlike the 3D transfer). To illustrate the efficiency of the proposed transformation, we show the cumulative density functions (cdf) of the three channels of CIE Lab color space and of the two components of the gradient fields ( $dx$  and  $dy$ ) for each result. The cdf of each result is compared against the cdf of the input and target images. For both results the input channels L, a and b have been transformed into the target ones with high precision. However, only the 5D transfer transforms the input gradient distribution into the target one. In contrast, the gradient cdf of the 3D result remains similar to the input gradient cdf.

Figure 6.7 demonstrates the efficiency of our 5D transformation. As illustrated by the plots in figure 6.7, our 3D MGGD-based method transforms the input color distribution without compromising the distribution of the input gradient field. In the 5D case, the color and gradient distributions are successfully transformed into the target ones. That is why, our 5D result has a higher contrast and it is much sharper than our 3D result.

Furthermore, figure 6.9 compares results from our 5D MGGD-based method with results from the 5D Monge-Kantorovich transformation. The latter assumes that the joint 5D color and gradient distributions of both the input and target images can be fitted by an MGD. However, the MGD fails to describe the sparse distributions of the input and target gradient fields, it also fails to carry out a joint transfer of color and gradient, as illustrated in figure 6.9.





**Figure 6.8** – Results of a color correction. Red boxes show the main differences (in color and gradient) between the corresponding result and the target image. The input and target images in the first row are extracted from the database in [93], whereas those in the second row are synthetic images from the same database. The results are best viewed on screen.

## 4.4 Color correction

Our method can also be employed to correct the color of images, used for image mosaicking and stitching. We use Xu and Mulligan’s database [93], from which we extract pairs of input and target images. The color of an extracted input image  $\mathbf{U}$  does not match the color of its corresponding target image  $\mathbf{V}$  and therefore, the input color needs to be corrected before stitching the images together. As the input and target images can differ not only in terms of color but also in terms of detail, we apply a joint 5D transfer of color and gradient instead of a 3D transfer of color. We apply our 5D simultaneous transfer between the input image  $\mathbf{U}$  and the target image  $\mathbf{V}$  (algorithm 4) and obtain a color corrected result  $\mathbf{F}$ , the joint color and gradient distribution of which is similar to the target distribution  $f_{\mathbf{V}}$ .

Figure 6.8 presents results from a color correction, obtained with our method as well as results from Reinhard *et al.*’s global parametric method [3] and the model-free method by Pitié *et al.* [60]. The results on the first row of the figure 6.8 show a good match in colors between our result and the target image (Reinhard’s and Pitié’s results do not match accurately the target color of both the Eiffel tower and the balloon). Red boxes visualize the main differences between the color of the results and the target color. Note the amount of blur in Pitié’s result as well as the artifacts on the top of the Eiffel tower.

The input and target images in the second row of figure 6.8 are synthetic images [93]. Given a target image  $\mathbf{V}$ , considered as the ground truth, the input image  $\mathbf{U}$  is obtained manually by modifying the colors of the image  $\mathbf{V}$ . Then, an automatic color correction method should recover the same target image  $\mathbf{V}$ .

All three results in the second row of figure 6.8 manage to correct the colors of the input image and to get close to the ground truth (the target image). However, there are subtle deviations from the target colors and gradient. For example, the color of the man’s clothes in Reinhard’s result is not an exact match of the target color (the orange color in Reinhard’s result appears darker than the one in the target image). Reinhard’s result is also less sharp than the target image (the red boxes). This is due to the fact that the input image is less sharp than the target image and the target details can only be recovered

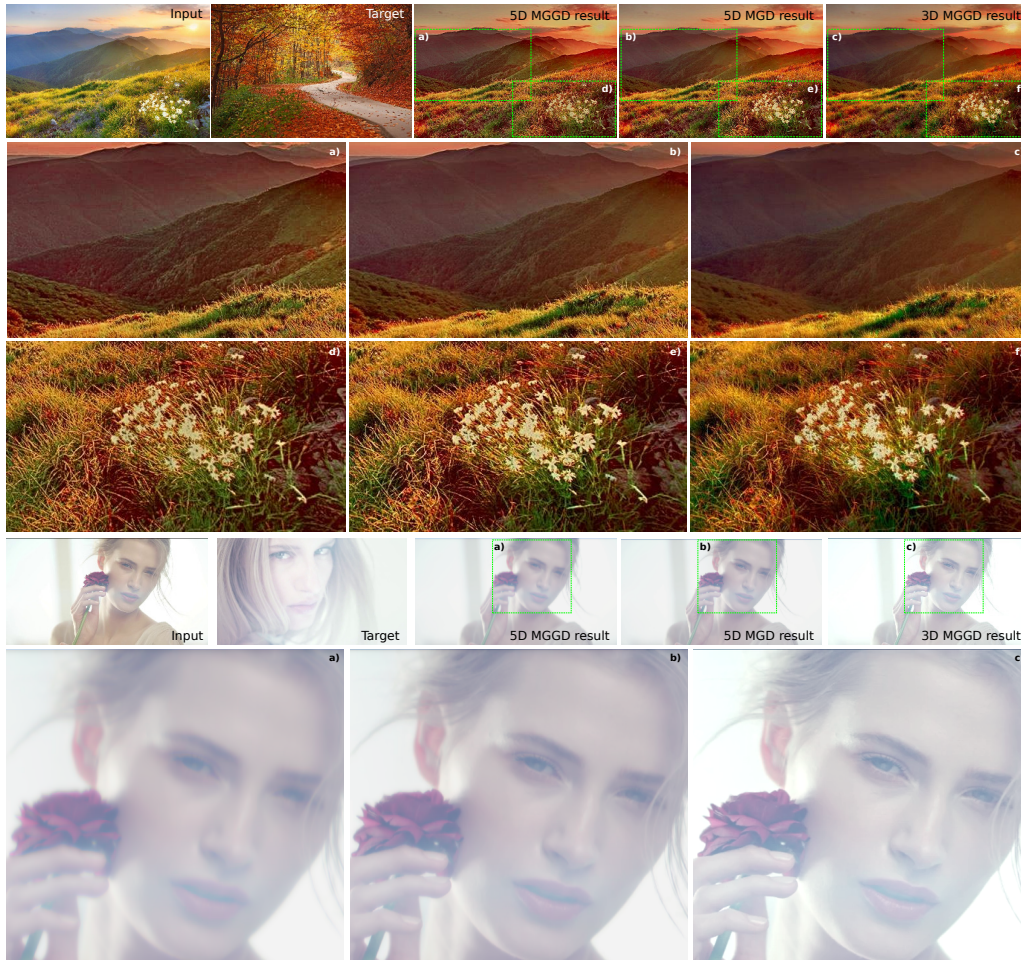
through edge sharpening (which Reinhard’s method does not perform). Furthermore, Pitié *et al.* [60] intentionally blur the result in order to remove artifacts, caused from the color transfer (de-graining). In contrast, our method restores the target colors and sharpens the details of the input image using a single 5D transformation of color and gradient.

---

## 5 Conclusion and future work

In this chapter, we have presented a novel transformation between input and target MGGDs. Two consequent steps transform the parameters of the input distribution according to the target distribution. The Monge-Kantorovich closed-form mapping transforms the scale, whereas the stochastic-based transfer transforms the shape of the input distribution. We have demonstrated the potential of our transformation in the context of a color transfer and a gradient transfer. We have also introduced a new simultaneous transfer of color and gradient, which can be applied for image color correction. The evaluation and the results, presented in the chapter, have illustrated the efficiency of our transformation for image editing applications. However, our method has some limitations. First, the proposed transformation cannot recover the smallest absolute gradient values and its efficiency is limited when applied to image de-blurring. Furthermore, the gradient transfer, proposed in this chapter, considers only the gradient distribution regardless of the gradient direction and structure. Therefore, the transfer of texture patterns is left for future work.

Furthermore, to further improve the color, light and gradient transfers, a mixture of MGGDs is to be considered in the future. It is indeed interesting to tackle distributions, which can be modelled by a mixture of MGGDs. If we carry out our global MGGD transformation for data, which cannot be fitted well enough by a single MGGD (*i.e.* a mixture of MGGDs is required), our method and Pitié *et al.*’s method [34] are very similar. For example, in the first row of figure 6.6, we observe that our method obtains a result similar to Pitié *et al.*’s result (though less-saturated and with less color artifacts). However, as we have shown in example 5 in figure 5.9 (chapter 5), those results can be improved with a mixture of Gaussian distributions (as the input and target images in the first row of figure 6.6 can be clustered according to their hue). Once the input and target images are clustered using a mixture of MGGDs (*i.e.* divided into clusters whose distributions follow an MGGD), the four mapping policies, presented in the previous chapter, can be employed to map the input and target clusters. Using a mixture of MGGDs has a great potential in the future for obtaining more natural, photo-realistic and better stylized images. In this chapter, we have showed that the color transfer can be improved using our MGGD-based transformation as the distributions of color and light in images are more accurately modelled by the MGGD than by the MGD. However, as the color, light and gradient distributions have bounded supports, unlike the MGGD, non-Gaussian distributions [94] would be more suitable to model color, light and gradient. In the following chapter, we exploit the benefits of using bounded distributions, and more specifically the Beta distribution, for color transfer.



**Figure 6.9** – Comparison between results of a simultaneous 5D MGGD-based transfer of color and gradient, 5D MGD-based transfer of color and gradient and 3D MGGD-based transfer of color. As demonstrated in figure 6.7, a 3D MGGD-based color transfer is insufficient to represent the “mood” of the target image, as it does not affect the input gradient field. The results in the forth row clearly show that a transfer of color is not sufficient to represent the style of an image. Both input and target images in the forth row are high-key images. However, they differ in the amount of details. Our 5D color and gradient transfer manages to smooth the details around the girl’s eyes, skin and rose with respect to the target image. Furthermore, a 5D transfer of color and gradient can also be performed using the Monge-Kantorovich transformation. However, as the latter fits the gradient distributions of both the input and target images by an MGD, it does not have a significant impact on the contrast and on the sharpness of the result (note that our 5D MGGD-based transfer enhances the best the details on the mountain slopes for the results in the first row). The visual comparison between the results is best observed on screen.



# Beta distribution transformation for color transfer 7

---

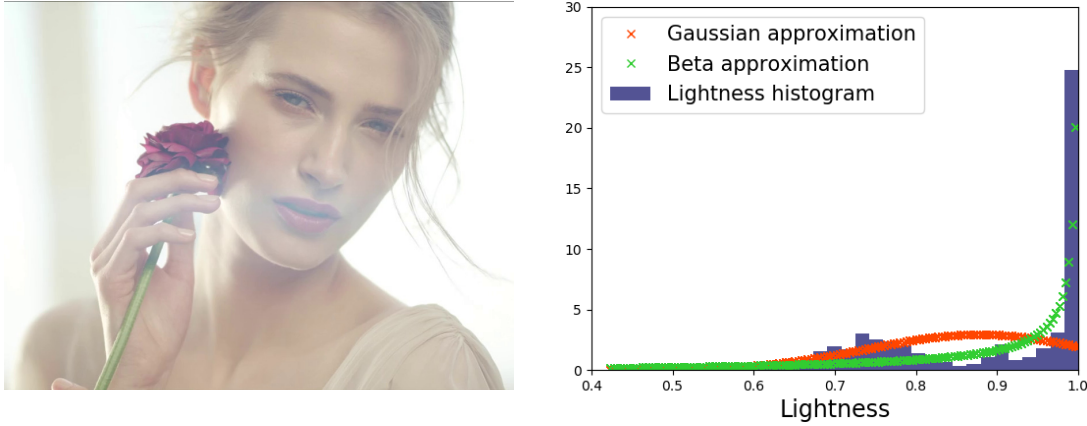
## 1 Introduction

In the previous two chapters, we have seen that color transfer is often viewed as a distribution transfer problem, in which the Gaussian distribution plays a significant role. Early research works on color transfer assume the color and light distributions of images follow a Gaussian distribution [3, 34], whereas local color transfer methods apply more precise models, such as Gaussian mixture models (GMM) [5–7].

So far, color transfer methods have been limited to Gaussian-based transformations, *e.g.* the color transformation, used in our style-aware color transfer method [7] (chapter 5). Moreover, in chapter 6, we have presented a new color transfer method, based on a transformation of the MGGD. The MGGD and in particular the MGD are continuous distributions with unbounded supports. Despite the fact that color and light in images are bounded in a finite interval, such as  $[0, 1]$ , they are still modelled by unbounded distributions, such as the MGGD and the MGD. When performing an MGGD-/MGD-based transformation between bounded distributions, we often obtain out-of-range values. Such values are cut off and eliminated, causing over-/under-saturation, out-of-gamut values, etc. To tackle the out-of-range limitation of the MGGD-/MGD-based transformations, in this chapter we adopt bounded distributions. Figure 7.1 illustrates an important limitation of the Gaussian approximation and presents the benefit of using a bounded Beta distribution to model color and light.

The Beta distribution is a bounded two-parameter-dependent distribution, which can admit different shapes and thus, can fit various data, bounded in a discrete interval. Adopting the Beta distribution to model color and light distributions of images is our key idea. In this chapter, we propose a novel transformation between two Beta distributions. Our transformation consists of four intermediate statistical transformations which progressively and accurately reshape an input Beta distribution into a target Beta distribution. We apply our Beta transformation both globally and locally in the context of a color transfer between images. Our results appear more natural and less saturated than results from recent state-of-the-art methods. Additionally, our results represent accurately the target color palette and truthfully portray the target contrast.

The rest of the chapter is organized as follows. Section 2 presents our Beta transformation. Results from applying the Beta transformation on images are shown in Section 3. The final section concludes the chapter.



**Figure 7.1** – The right-hand plot illustrates the lightness histogram of the left-hand image as well as two approximations, namely Gaussian and Beta. The Gaussian distribution provides a poor approximation of the asymmetric lightness distribution. In contrast, the Beta distribution models the image lightness more accurately by accounting for its right-skewness.

## 2 Beta transformation

The following section consists of two parts. In the first part we present the Beta distribution, whereas in the second part we introduce our Beta transformation.

### 2.1 Beta distribution

Hereafter, we present several well-known statistical transformations which play an important role in our Beta transformation.

#### 2.1.1 Beta-Fisher relationship

Let  $\mathbf{x} \sim \text{Beta}(\alpha, \beta)$ . Then, a variable  $\mathbf{y}$ , obtained as  $\mathbf{y} = f_{BF}(\mathbf{x}, \alpha, \beta)$ , where function  $f_{BF}(\cdot)$  is defined as follows:

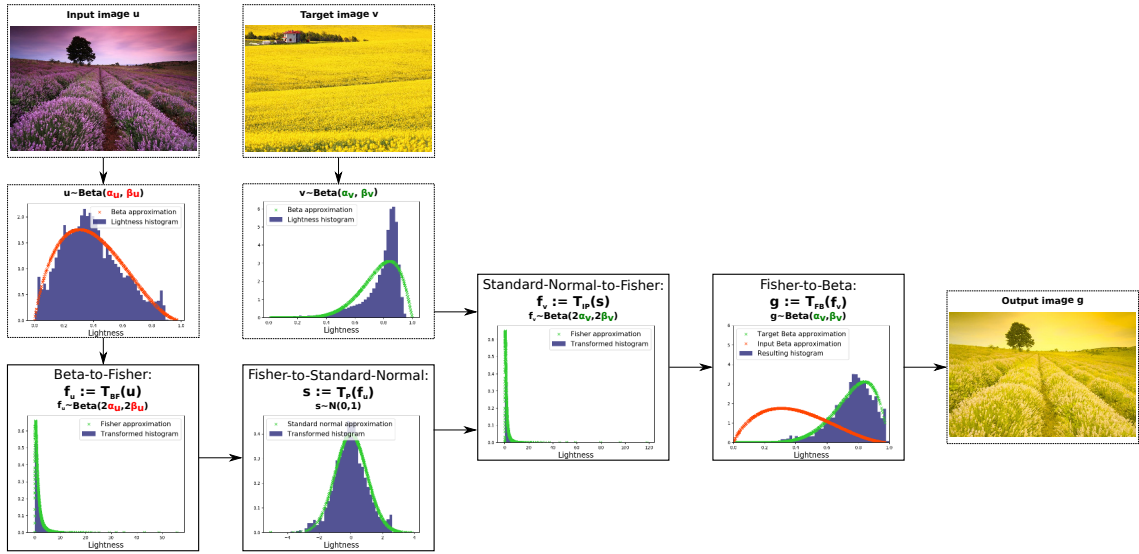
$$f_{BF}(\mathbf{x}, \alpha, \beta) = \frac{\beta \mathbf{x}}{\alpha(1 - \mathbf{x})}, \quad (7.1)$$

is a Fisher variable with shape parameters  $2\alpha$  and  $2\beta$  (denoted  $\mathbf{y} \sim \mathbf{F}(2\alpha, 2\beta)$ ). Equation (7.1) maps the bounded interval  $[0, 1]$  into the semi-bounded interval  $[0, \infty)$  with  $\lim_{\mathbf{x} \rightarrow 1} \mathbf{y} = \infty$ .

Reversely, a variable  $\mathbf{z} = f_{FB}(\mathbf{y}, \alpha, \beta)$ , where  $\mathbf{y} = f_{BF}(\mathbf{x}, \alpha, \beta)$  and  $f_{FB}(\cdot)$  is obtained as follows:

$$f_{FB}(\mathbf{y}, \alpha, \beta) = \frac{\alpha \mathbf{y}}{\beta + \alpha \mathbf{y}}, \quad (7.2)$$

is a Beta variable, e.g.  $\mathbf{z} \sim \text{Beta}(\alpha, \beta)$ . Equation (7.2) maps the semi-bounded interval  $[0, \infty)$  into the bounded interval  $[0, 1]$  with  $\lim_{\mathbf{y} \rightarrow \infty} \mathbf{z} = 1$ .



**Figure 7.2** – Our Beta transformation consists of four intermediate transformations (steps). The first two steps progressively transform the input distribution into a standard normal distribution. Then, using the target distribution, the last two transformations reshape the standard normal distribution into a Beta distribution with shape parameters, close to the target shape parameters. For each step of our method, the flowchart shows the transformed distribution and its corresponding approximation. The input and target distributions are extracted from image lightness channels.

### 2.1.2 Fisher-Chi-square relationship

Let  $\mathbf{x}$  and  $\mathbf{y}$  be two Chi-square random variables with  $\alpha$  and  $\beta$  degrees of freedom respectively ( $\mathbf{x} \sim \chi_\alpha^2, \mathbf{y} \sim \chi_\beta^2$ ). Then:

$$\frac{\mathbf{x}/\alpha}{\mathbf{y}/\beta} \sim \mathbf{F}(\alpha, \beta) \quad (7.3)$$

## 2.2 Transformation between Beta distributions

Hereafter, we present our 1D Beta transformation. Let  $\mathbf{u}$  and  $\mathbf{v}$  be 1D input and target random variables, following a Beta distribution, *i.e.*  $\mathbf{u} \sim \text{Beta}(\alpha_{\mathbf{u}}, \beta_{\mathbf{u}})$  and  $\mathbf{v} \sim \text{Beta}(\alpha_{\mathbf{v}}, \beta_{\mathbf{v}})$ . We aim to transform the distribution of  $\mathbf{u}$  into a distribution, similar to the target distribution. Transforming one Beta distribution into another one in a single pass could be challenging. That is why, our transformation consists of four intermediate transformations which are based on known approximations:

1. Transformation of  $\mathbf{u}$  into  $\mathbf{f}_{\mathbf{u}} \sim \mathbf{F}(2\alpha_{\mathbf{u}}, 2\beta_{\mathbf{u}})$ ;
2. Transformation of  $\mathbf{f}_{\mathbf{u}}$  into a standard normal variable  $\mathbf{s}_{\mathbf{u}}$ ;
3. Transformation of  $\mathbf{s}_{\mathbf{u}}$  into  $\mathbf{f}_{\mathbf{v}} \sim \mathbf{F}(2\alpha_{\mathbf{v}}, 2\beta_{\mathbf{v}})$ ;
4. Transformation of  $\mathbf{f}_{\mathbf{v}}$  into  $\mathbf{g} \sim \text{Beta}(\alpha_{\mathbf{v}}, \beta_{\mathbf{v}})$ .

The steps of our transformation are illustrated in figure 7.2.

### 2.2.1 Beta-to-Fisher

We transform the input Beta variable  $\mathbf{u}$  into a Fisher variable  $\mathbf{f}_u = \mathbf{T}_{BF}(\mathbf{u})$  with shape parameters  $2\alpha_u$  and  $2\beta_u$  as follows:

$$\mathbf{T}_{BF} : \mathbf{u} \rightarrow f_{BF}(\mathbf{u}, \alpha_u, \beta_u), \quad (7.4)$$

where function  $f_{BF}(\cdot)$  is defined in (7.1). Once we obtain the Fisher variable  $\mathbf{f}_u$ , we transform it into standard normal variable  $\mathbf{s}$  in the second step of our transformation.

### 2.2.2 Fisher-to-Standard-Normal

The transformation of  $\mathbf{f}_u$  is based on Paulson's equation [95, 96]. Hereafter, we derive Paulson's equation using Fieller's approximation [97].

Fieller's approximation transforms the ratio of two normally distributed variables into a standard normal variable. Let  $\mathbf{x} \sim \mathcal{N}(\mu_x, \sigma_x^2)$  and  $\mathbf{y} \sim \mathcal{N}(\mu_y, \sigma_y^2)$  be two normally distributed random variables. Then, Fieller approximation [97] admits the following form:

$$f_{FL}(\mathbf{x}, \mathbf{y}, \mu_x, \mu_y, \sigma_x, \sigma_y) = \frac{\frac{\mathbf{x}}{\mathbf{y}}\mu_y - \mu_x}{\sqrt{\left(\frac{\mathbf{x}}{\mathbf{y}}\right)^2 \sigma_y^2 + \sigma_x^2}} \quad (7.5)$$

Each variable  $\mathbf{s}$ , where  $\mathbf{s} = f_{FL}(\mathbf{x}, \mathbf{y}, \mu_x, \mu_y, \sigma_x, \sigma_y)$ , is a standard normal variable.

Paulson's equation can be derived from (7.5) by computing the normal variables  $\mathbf{x}$  and  $\mathbf{y}$  using a Chi-square distribution. Let  $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$  be a normal variable and  $\mathbf{w} \sim \chi_\gamma^2$  be a Chi-square variable with  $\gamma$  degrees of freedom. Then,  $\mathbf{z}$  can be expressed as follows [98]:

$$\mathbf{z} = \left( \frac{\mathbf{w}}{\gamma} \right)^{\frac{1}{p}}, \quad (7.6)$$

where  $p \in \mathbb{N}$  and  $\mu$  and  $\sigma$  are functions of  $\gamma$  and  $p$ :

$$\sigma^2 = f_\sigma(\gamma, p) = \frac{2}{\gamma p^2} \text{ and } \mu = f_\mu(\sigma) = 1 - \sigma^2. \quad (7.7)$$

Using (7.6), we express the normal variables  $\mathbf{x}$  and  $\mathbf{y}$  in (7.5) as follows:  $\mathbf{x} = (\mathbf{w}_x/\alpha)^{\frac{1}{p}}$  and  $\mathbf{y} = (\mathbf{w}_y/\beta)^{\frac{1}{p}}$ , where  $\mathbf{w}_x \sim \chi_\alpha^2$  and  $\mathbf{w}_y \sim \chi_\beta^2$ , and  $p \in \mathbb{N}$ . Let  $\mathbf{f} = \frac{\mathbf{x}^p}{\mathbf{y}^p} = \frac{\mathbf{w}_x/\alpha}{\mathbf{w}_y/\beta}$ . Then, Paulson's equation transforms the variable  $\mathbf{f}$ , into a standard normal variable  $\mathbf{s}$  as follows [95, 96]:

$$f_P(\mathbf{f}, \mu_x, \mu_y, \sigma_x, \sigma_y, p) = \frac{\mathbf{f}^{\frac{1}{p}}\mu_y - \mu_x}{\sqrt{\mathbf{f}^{\frac{2}{p}}\sigma_y^2 + \sigma_x^2}}, \quad (7.8)$$

where  $\mathbf{s} = f_P(\mathbf{f}, \mu_x, \mu_y, \sigma_x, \sigma_y, p)$ ,  $\sigma_x^2 = f_\sigma(\alpha, p)$ ,  $\sigma_y^2 = f_\sigma(\beta, p)$ ,  $\mu_x = f_\mu(\sigma_x)$  and  $\mu_y = f_\mu(\sigma_y)$  (from (7.7)). From (7.3), it becomes clear that  $\mathbf{f}$  is a Fisher variable with shape parameters  $\alpha$  and  $\beta$ , i.e.  $\mathbf{f} \sim \mathbf{F}(\alpha, \beta)$ . Therefore, Paulson's equation transforms a Fisher variable into a standard normal variable.

We adopt Paulson's equation (7.8) to transform the Fisher variable  $\mathbf{f}_u \sim \mathbf{F}(2\alpha_u, 2\beta_u)$  (computed with (7.4)) into a standard normal variable  $\mathbf{s}_u = \mathbf{T}_P(\mathbf{f}_u)$ . The transformation  $\mathbf{T}_P$  is defined as follows:

$$\mathbf{T}_P : \mathbf{f}_u \rightarrow f_P(\mathbf{f}_u, \mu_\alpha^u, \mu_\beta^u, \sigma_\alpha^u, \sigma_\beta^u, p), \quad (7.9)$$

where  $p \in \mathbb{N}$  and  $(\sigma_\alpha^u)^2 = f_\sigma(2\alpha_u, p)$ ,  $(\sigma_\beta^u)^2 = f_\sigma(2\beta_u, p)$ ,  $\mu_\alpha^u = f_\mu(\sigma_\alpha^u)$  and  $\mu_\beta^u = f_\mu(\sigma_\beta^u)$ .

### 2.2.3 Standard-Normal-to-Fisher

Once we compute the standard normal variable  $\mathbf{s}_u$ , we inverse Paulson's equation (7.8) to transform  $\mathbf{s}_u$  into a Fisher variable  $\mathbf{f}_v$ . We carry out the inversed Paulson's equation using the target shape parameters  $\alpha_v$  and  $\beta_v$  instead of the input shape parameters  $\alpha_u$  and  $\beta_u$ .

Let  $\alpha$  and  $\beta$  be any two shape parameters. We first present the inversed Paulson's equation for transforming any standard normal variable  $\mathbf{s}$  into a Fisher variable  $\mathbf{f} \sim \mathbf{F}(2\alpha, 2\beta)$ :

$$(\mathbf{s}^2 \sigma_y^2 - \mu_y^2) \mathbf{f}^{\frac{1}{p}} + 2\mu_x \mu_y \mathbf{f}^{\frac{1}{p}} + \mathbf{s}^2 \sigma_x^2 - \mu_x^2 = 0, \quad (7.10)$$

where  $\sigma_x^2 = f_\sigma(2\alpha, p)$ ,  $\sigma_y^2 = f_\sigma(2\beta, p)$ ,  $\mu_x = f_\mu(\sigma_x)$  and  $\mu_y = f_\mu(\sigma_y)$ . The inversed Paulson's equation (7.10) can be solved as a quadratic equation for  $\mathbf{t} = \mathbf{f}^{\frac{1}{p}}$ . Let  $\mathbf{t} = (\mathbf{t}_1, \dots, \mathbf{t}_n)$  and  $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n)$ , where  $\mathbf{t}_i$  and  $\mathbf{s}_i$  are the samples  $\mathbf{t}$  and  $\mathbf{s}$  respectively. Then, the two solutions  $\mathbf{t}_1$  and  $\mathbf{t}_2$  of (7.10) are computed  $\forall i \in \{1, \dots, n\}$  using a function  $f_{IP}(\mathbf{s}_i, \mu_x, \mu_y, \sigma_x, \sigma_y, C)$  (Ashby [96] presents the solutions for  $p = 1/3$ ):

$$\mathbf{f}_1^{\frac{1}{p}} = \mathbf{t}_1 = f_{IP}(\mathbf{s}_i, \mu_x, \mu_y, \sigma_x, \sigma_y, 1), \quad (7.11)$$

$$\mathbf{f}_2^{\frac{1}{p}} = \mathbf{t}_2 = f_{IP}(\mathbf{s}_i, \mu_x, \mu_y, \sigma_x, \sigma_y, -1), \text{ where} \quad (7.12)$$

$$f_{IP}(\mathbf{s}_i, \mu_x, \mu_y, \sigma_x, \sigma_y, C) = \frac{-\mu_x \mu_y + C\sqrt{D}}{\mathbf{s}_i^2 \sigma_y^2 - \mu_y^2} \quad (7.13)$$

and  $D = \mathbf{s}_i^2(\sigma_x^2 + \sigma_y^2 + \sigma_x^2 \sigma_y^2(\sigma_x^2 + \sigma_y^2 - \mathbf{s}_i^2 - 4))$ , and  $C = \pm 1$ .

Now, we solve the inversed Paulson's equation (7.10) for the target shape parameters  $\alpha_v$  and  $\beta_v$ . In (7.13), we replace  $\mathbf{s}$  by  $\mathbf{s}_u$  (computed with (7.9)) and the functions  $\sigma_x, \sigma_y, \mu_x$  and  $\mu_y$  by the following functions respectively:  $(\sigma_\alpha^v)^2 = f_\sigma(2\alpha_v, p)$ ,  $(\sigma_\beta^v)^2 = f_\sigma(2\beta_v, p)$ ,  $\mu_\alpha^v = f_\mu(\sigma_\alpha^v)$  and  $\mu_\beta^v = f_\mu(\sigma_\beta^v)$ . That way, we obtain a Fisher variable  $\mathbf{f}_v \sim \mathbf{F}(2\alpha_v, 2\beta_v)$ . Each sample  $\mathbf{f}_v^i$  of  $\mathbf{f}_v$  is computed using a transformation  $\mathbf{T}_{IP}$ , i.e.  $\mathbf{f}_v^i = \mathbf{T}_{IP}(\mathbf{s}_u^i) \forall i \in \{1, \dots, n\}$ , where  $\mathbf{T}_{IP}$  is defined as follows:

$$\mathbf{T}_{IP} : \begin{cases} \mathbf{s}_u^i \rightarrow \left( f_{IP}(\mathbf{s}_i, \mu_\alpha^v, \mu_\beta^v, \sigma_\alpha^v, \sigma_\beta^v, 1) \right)^p, & \text{if } \mathbf{s}_u^i < 0, \\ \mathbf{s}_u^i \rightarrow \left( f_{IP}(\mathbf{s}_i, \mu_\alpha^v, \mu_\beta^v, \sigma_\alpha^v, \sigma_\beta^v, -1) \right)^p, & \text{if } \mathbf{s}_u^i \geq 0. \end{cases} \quad (7.14)$$

### 2.2.4 Fisher-to-Beta

In the final step of our transformation, we transform the Fisher variable  $\mathbf{f}_v$  into a Beta variable  $\mathbf{g}$  using a transformation  $\mathbf{T}_{FB}$ :

$$\mathbf{T}_{FB} : \mathbf{f}_v \rightarrow f_{FB}(\mathbf{f}_v, \alpha_v, \beta_v), \quad (7.15)$$

where function  $\mathbf{f}_{FB}(\cdot)$  is defined in (7.2). The variable  $\mathbf{g} = \mathbf{T}_{FB}(\mathbf{f}_v)$  is approximately distributed according to a Beta distribution with shape parameters  $\alpha_v$  and  $\beta_v$ , i.e. its distribution is similar to the target distribution.



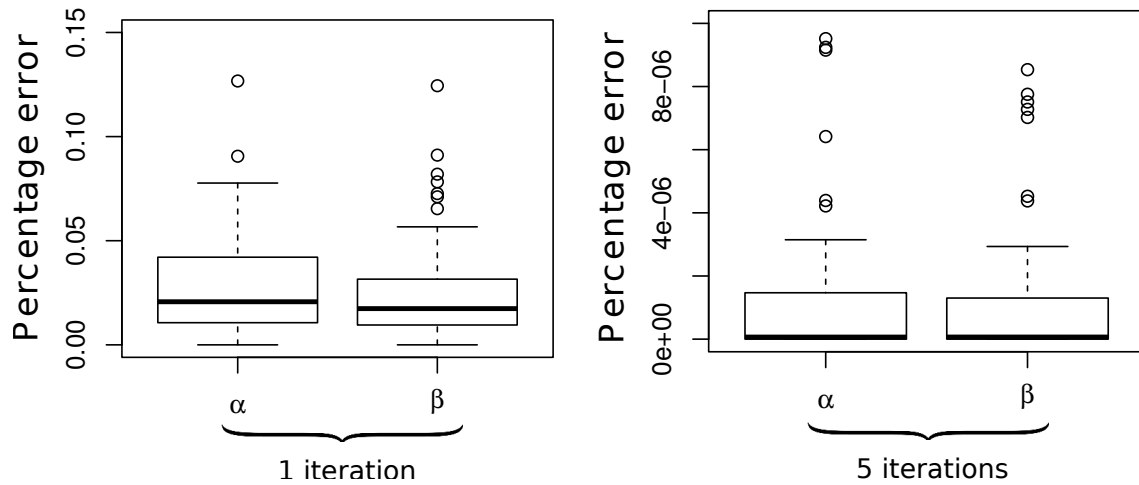
## 2.2.5 Choice of $p$

The solutions  $\mathbf{t}_1$  and  $\mathbf{t}_2$  of (7.10) can be both negative and positive, depending on  $\mu_\alpha^v$ ,  $\mu_\beta^v$ ,  $\sigma_\alpha^v$  and  $\sigma_\beta^v$ . In contrast, the values of the variables, distributed according to the Fisher law, are non-negative. To this end, we choose  $p = 4$ , following Hawkins et al.'s proposition [98]. By choosing  $p = 4$ , we make sure that Paulson's equation (7.8) holds for small values of the shape parameters  $\alpha_u$  and  $\beta_u$  [98]. That way, we also ensure that each component  $\mathbf{f}_v^i$  of  $\mathbf{f}_v$  is non-negative (see (7.14)).

## 2.3 Evaluation of Beta transformation

To evaluate the performance of our Beta transformation, we carry out the transformation on 111 pairs of input and target Beta distributions. Then, we compute the percentage error between the shape parameters of each resulting (from the transformation) distribution and the target shape parameters. The percentage error distributions (obtained after 1 and 5 iterations of our transformation) are shown in figure 7.3. After 1 iteration, the mean percentage error is less than 0.05 and it converges towards 0 after 5 iterations.

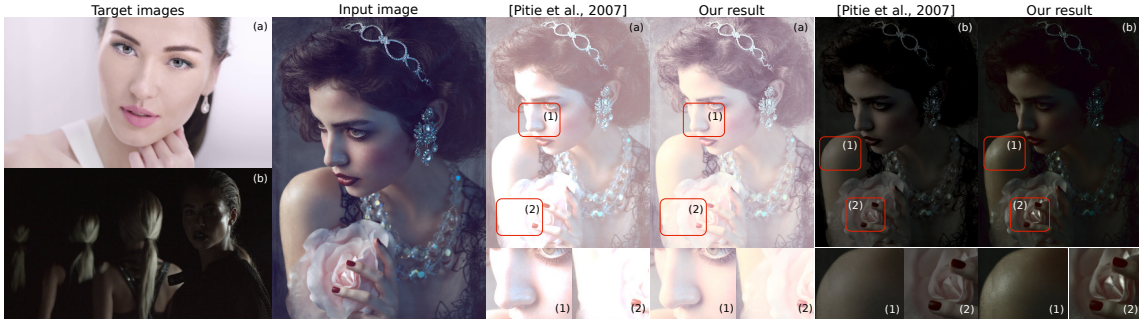
The plots in figure 7.3 illustrate the benefit of performing our Beta transformation iteratively. After a single iteration, the percentage error is already significantly small. The more we iterate, the smaller the percentage error. For the application, shown in this chapter (*i.e.* color transfer), our experiments indicate that the change in the resulting distribution becomes negligible after the fifth iteration. Therefore, the results, shown in this chapter, are obtained using five iterations of our Beta transformation.



**Figure 7.3** – Box-and-Whisker plots of the percentage errors for the two Beta shape parameters (computed after 1 and 5 iterations of our transformation).

## 3 Results

We apply our Beta transformation in the context of a color transfer between input and target images. We carry out our 1D Beta transformation independently on each pair of



**Figure 7.4** – Global color transfer. Our results portray the contrast of the target image better than the Pitié et al.’s results and they appear sharper than Pitié et al.’s results. Snippets (1) and (2) illustrate differences between our results and Pitié et al.’s results.

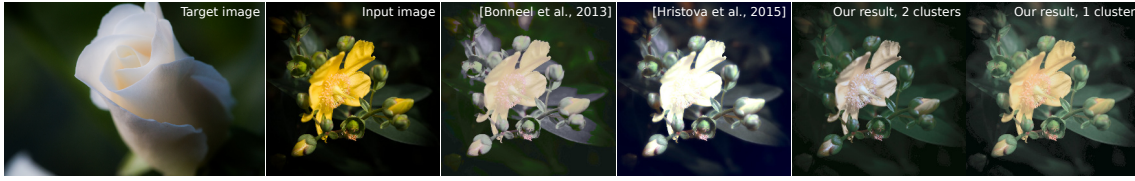
input/target color channels. To this end, we use CIE Lab, as the color space provides efficient channel decorrelation. Each of our results in this chapter is obtained for 5 iterations of our transformation. We perform the Beta transformation globally as well as locally.

### 3.1 Global color transfer

To perform a global color transfer, we first model the distribution of each input/target channel by a Beta distribution and then, we carry out our Beta transformation between the channel distributions. In figure 7.4, our global Beta-based color transfer is compared to the Gaussian-based 3D linear transformation by Pitié et al. [34] (also carried out in CIE Lab). The target image (a) is characterized by a low contrast and an absence of strongly defined highlights. Like the target image (a), our result (a) does not contain highly contrasting regions. In contrast, Pitié et al.’s result (a) contains regions of well-defined highlights, appearing on the girl’s face, shoulder and flower. Such highlights are not present in the target image (a) and make the contrast of Pitié et al.’s result non-uniform. Furthermore, the target image (b) in figure 7.4 is characterized by a presence of strong highlights and deep shadows. The highly contrasting regions of the target image (b) are well-represented in our result (b), whereas the absence of strong highlights lowers the contrast of Pitié et al.’s result (b).

### 3.2 Local color transfer

Instead of modeling the entire distributions of color and light by Beta distributions, we can build a more accurate model using Beta mixture models (BMM) [4]. We use BMM to cluster the input and target images according to one of two components, *i.e.* lightness or hue. We adopt the classification method, proposed in chapter 5, to determine the best clustering component for each image. Additionally, we let clusters overlap and we use the BMM soft segmentation to compute the overlapping pixels. Then, we apply one of the four mapping policies [7], proposed in chapter 5, to map the input and target clusters and we carry out our Beta transformation between each pair of corresponding clusters. The overlapping pixels are influenced by more than one Beta transformation. To determine the final value of an overlapping pixel, we compute the average of all transformations, containing the pixel, using exponential decay weights.

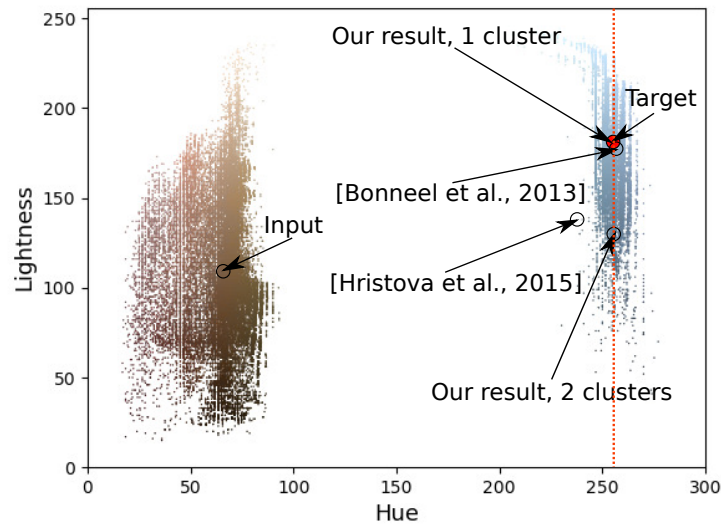


**Figure 7.5** – The local Beta transformation is more efficient than the global Beta transformation in cases when the input/target color and light distributions cannot be well-modelled by a single Beta distribution. We use 2 clusters to obtain a naturally-looking result, matching accurately the target color palette and contrast.

Figure 7.5 illustrates the benefit of applying a local color transfer between images. A global Beta transformation fails to transfer properly the target color palette to the input image. This is due to the fact that the input and target color distributions cannot be modelled well enough by a single Beta distribution. To improve the result from the global transfer, we apply our Beta transformation locally, using 2 clusters. That way, BMM, as more precise distribution models for the input/target distributions, help transfer the target colors more accurately. We compare our local method with Bonneel et al.’s [6] local method and our style-aware local method [7], both of which carry out Gaussian-based transformations. Bonneel et al.’s method fails to match the target floral color. Our style-aware local method transfers correctly the background and foreground colors but it results in an overexposure of the foreground pixels. This overexposure is partially due to the local CAT in our style-aware method, which, in this case, overadapts the colors of the result to the target illuminant (note that the target image is more illuminated than the input image). The presence of overexposed pixels in the result compromises its photo-realism. In contrast, the result, obtained locally with our Beta transformation, better portrays the natural cream white color of the target rose without overexposing it.

Furthermore, figure 7.7 shows the impact of our local Beta-based color transfer on the input contrast preservation. We observe that when we apply our Beta transformation globally, we accurately transfer the target color palette. However, we also decrease the contrast of the input image. Due to the specific contents of the input/target images, users may expect a less overexposed result with a higher contrast (like the input image). To preserve the input contrast, we partition the input and target images in figure 7.7 into two clusters, *i.e.* highlights and shadows. We map the target to the input clusters using our Light to Light mapping policy (presented in chapter 5). Then, we perform our Beta transformation between each two corresponding clusters and obtain the final result. That way, we manage to transfer the target colors without compromising the input contrast (as shown in figure 7.7).

As the input and target images in figure 7.7 consist of a single dominant color, the average hue provides a decent statistic for measuring the similarity of each result to the target color palette. Figure 7.6 presents a plot of the lightness-hue distributions of the input and target images from figure 7.7. The mean lightness and the mean hue of each result in figure 7.7 are displayed in circles in figure 7.7. Our result, obtained using our Beta distribution with two clusters, has the same mean hue as the target image. In contrast, the result, obtained with our style-aware local method (again using two luminance clusters), has an out-of-gamut mean hue. Indeed, a closer visual comparison of both of our local results (figure 7.7) indicates that using our Beta transformation improves the similarity



**Figure 7.6** – Lightness-hue distribution plot of the input and target images from figure 7.7. The left cluster corresponds to the color distribution of the input image, whereas, the right cluster corresponds to the color distribution of the target image. The red circle illustrates the mean target lightness and hue, whereas the dashed red line visualizes the mean target hue.

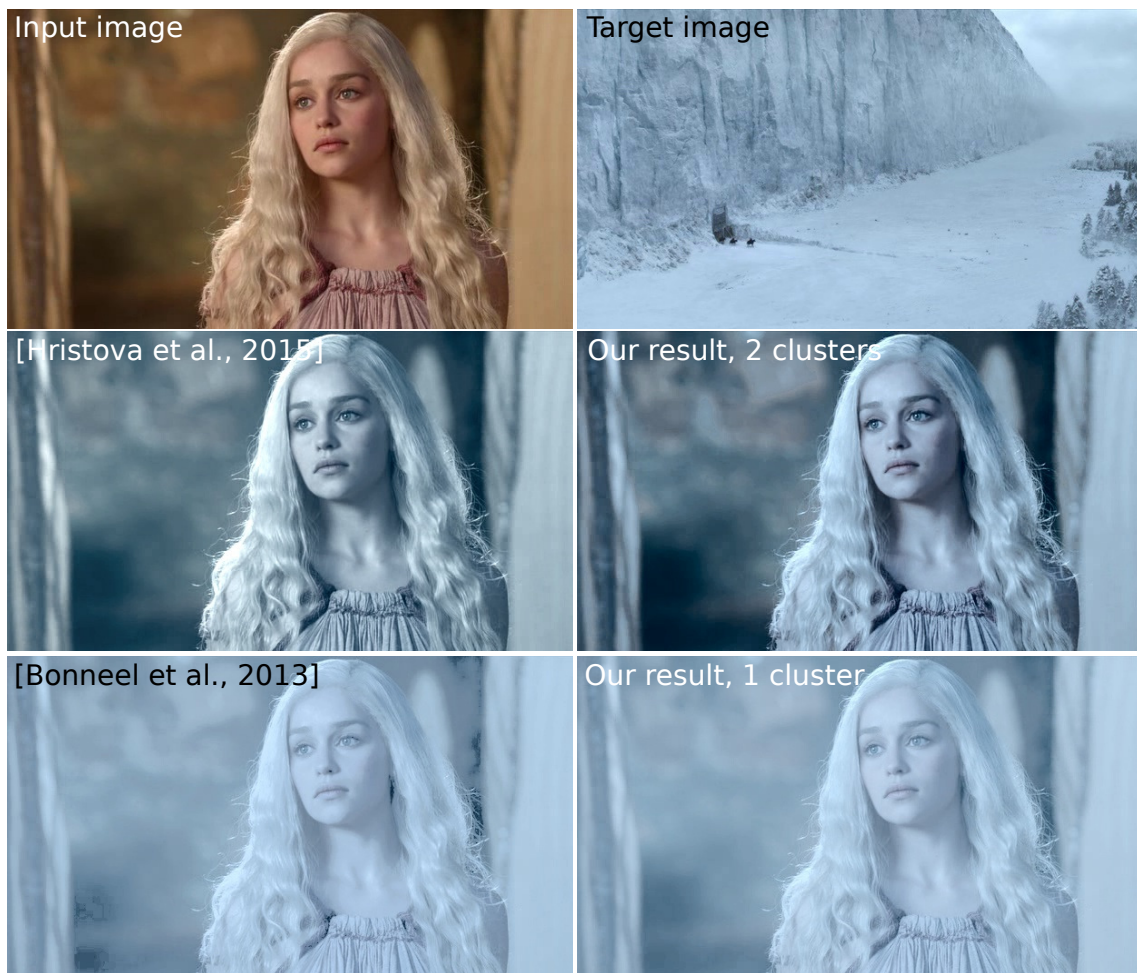
between the target color palette and the colors of our result. The hue statistic shows that our local result (obtained with our Beta transformation) matches the target colors better than the result, obtained with our style-aware method. Furthermore, the difference in the mean lightness between each of our local results and the target image can be explained by the desired preservation of the input contrast. More results, obtained with our local Beta transformation, are shown in figure 7.8.

## 4 Conclusion

In this chapter, we have presented a transformation between two Beta distributions. Our main idea involved modelling color and light image distributions using Beta distributions (or BMM) as an alternative to the Gaussian distribution (or GMM). We have applied our Beta transformation in the context of a color transfer between images, though the transformation can be applied to any bounded data, following the Beta distribution law. Our results have shown the great efficiency of our method for transferring the target colors and contrast. The potential of our Beta transformation can further be extended to high-dynamic-range imaging and video sequences, which are considered as future venues for improvement.

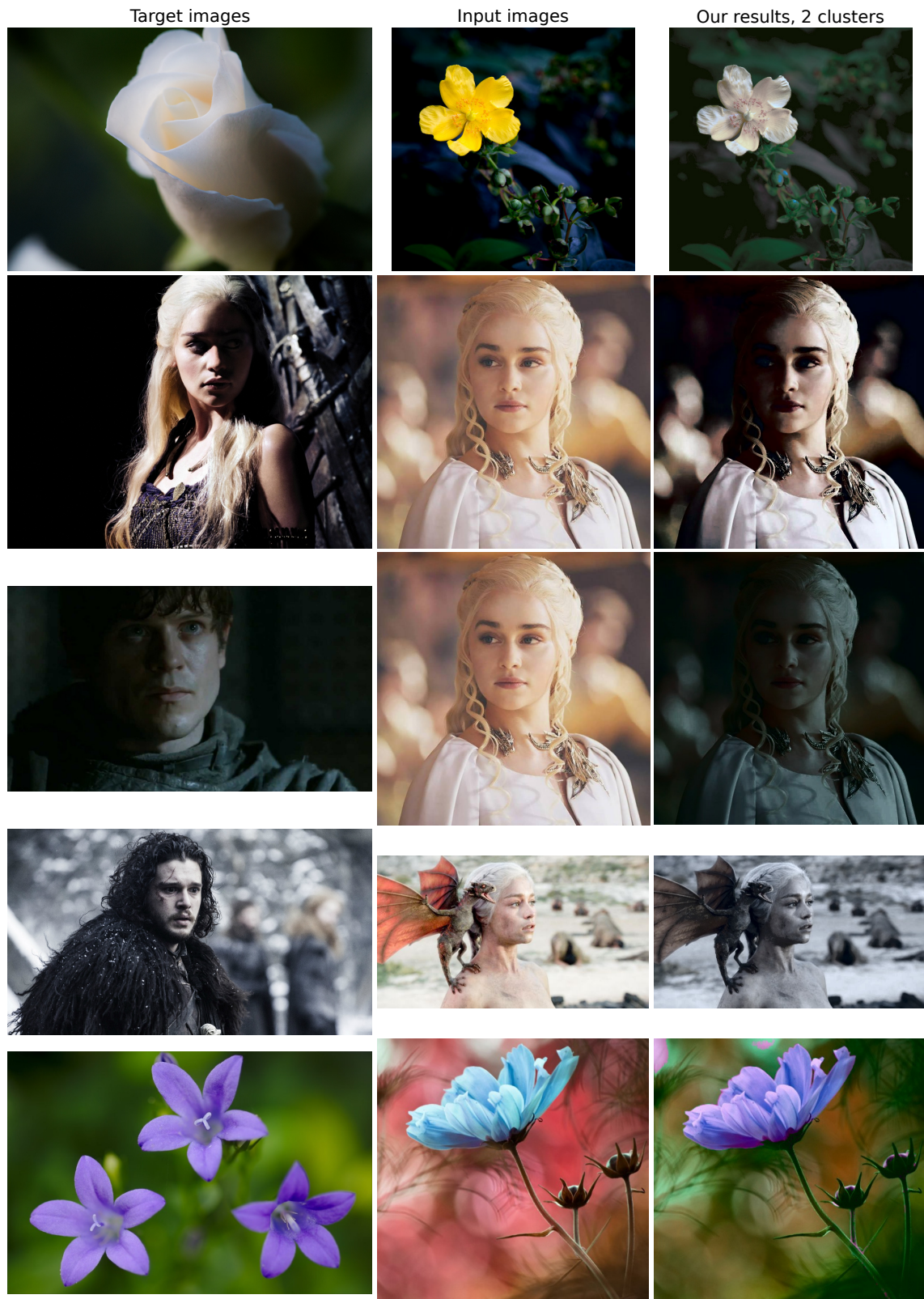
In the present and the two previous chapters, we have introduced three statistical transformations, applied mainly in the context of color and style transfers. We have carried out both objective and subjective evaluations to compare our results to results from state-of-the-art methods. The process of evaluating the performance of a color transfer may be time-consuming and may introduce a certain bias. The following chapter addresses the performance assessment of color transfer methods. The model, presented in the following





**Figure 7.7** – Our local color transfer may significantly influence the contrast of the resulting image. Our global method accurately transfers the target color palette, but it also decreases the contrast, resulting in a non-natural flat image. On the other hand, our local Beta transformation preserves the input contrast.

chapter, connects the subjective judgment on the quality of a color transfer to a set of objective metrics. The model proves to be an efficient and practical tool for assessing the quality of a color transfer method.



**Figure 7.8** – Additional results of our Beta-based local color transfer. The results are obtained using 2 clusters and our mapping policies, introduced in chapter 5.



# Perceptual metric for color transfer methods 8

---

## 1 Introduction

As presented in the previous three chapters, different color transfer methods often result in different output images. The process of determining the most plausible output image may be subjective, as it depends on a person's preference. Due to the lack of an objective metric for evaluating results from a color transfer, comparisons between existing methods are often carried out through a user study. Conducting a user study for each newly proposed method may be a tedious and time-consuming task. Moreover, the conditions, under which the study is conducted, and its protocol may vary with the group of people handling the evaluation process. That makes the comparison between different methods and the assessment of their performance challenging.

To ease the evaluation process, in this chapter, we propose a model for objective evaluation of the color transfer quality. Our model explains the relationship between users' perception and a number of image features. To account for users' perception, we first conduct a user study on various color transfer results from six state-of-the-art methods. The study combines the aesthetic quality of the result with the quality of the color transfer, as perceived by the users. Then, for each result, we compute nine image features, which objectively describe the quality of a color transfer. We use these features to predict the scores from our user study.

We fit the set of the subjective scores and image features by a regression model with random forests. Our analysis shows that a regression with random forests is more accurate than linear and non-linear regressions. Our model is a general tool for assessing the perceptual performance of a color transfer. To this end, our model introduces an objective metric between three images - the input, the target and the result.

The chapter is organized as follows. Section 2 presents commonly used metrics for objective image evaluation. Our user study and our regression model are introduced in section 3.2. An analysis of the proposed model is carried out in section 4. Finally, the last section concludes the chapter.

**Publication.** The work, presented in this chapter, has been published in the following paper:

*H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, "Perceptual metric for color transfer methods", in IEEE International Conference on Image Processing, 2017.*



## 2 Related works

As discussed in chapter 5, the performance evaluation of a color transfer is often addressed from two main perspectives [7]. First, a good color transfer method should not compromise the quality of the input image. Second, it should ensure a good transfer of color from the target image to the result. Hereafter, we present objective metrics, commonly used for color transfer evaluation.

**Structural similarity metric (SSIM).** This full-reference metric [68] measures the perceived quality of an image with regards to the original distortion-free image. In the color transfer context, it is used as a similarity metric between the input image and the result to measure the degree of artifacts in the result [7, 20]. To this end, SSIM is applied on the luminance channel of both the input and the resulting images.

**Peak signal-to-noise ratio (PSNR).** Like SSIM, PSNR is used as a quality measurement between an original image and a compressed one. And while SSIM outperforms PSNR as a similarity measure, PSNR has been adopted in the context of color dissimilarity by Hwang et al. [44]. The authors evaluate their color transfer method by measuring the difference in color (in terms of PSNR) between the input image and the result. The ultimate goal of Hwang et al.’s method is to maximize the PSNR metric.

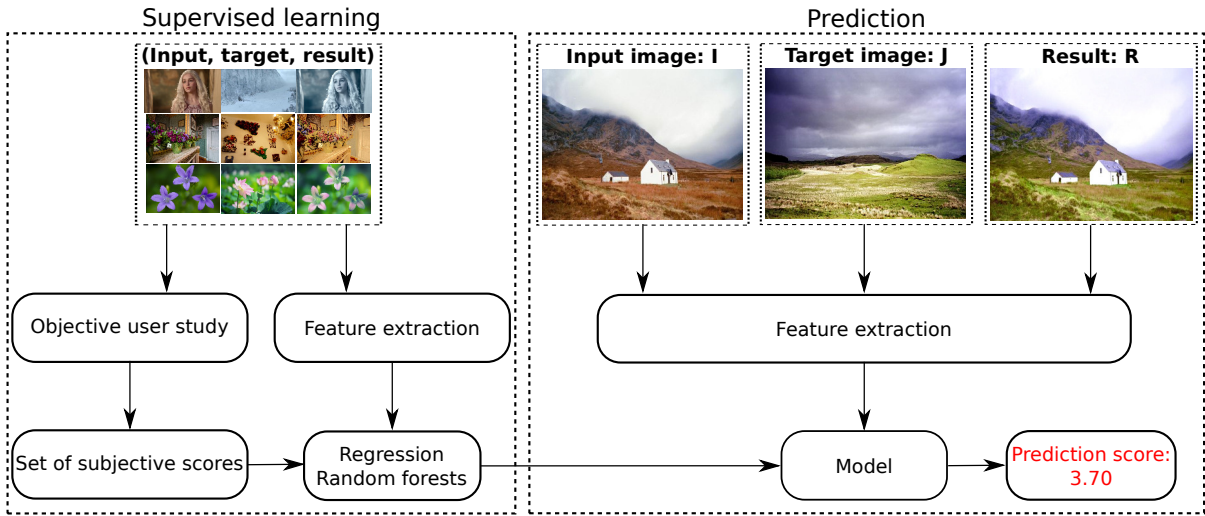
**Bhattacharya coefficient.** The metric measures the amount of overlap between two distributions [70]. In chapter 5, we have adopted it to compute the color similarity between a result from our style-aware color transfer and a target image [7].

**Out-of-gamut metric.** The metric has been proposed by Nguyen et al. [43] and used to evaluate their gamut-based color transfer method. The metric computes the distance between the gamut of the resulting image and the gamut of the target image.

The use of a combination of objective metrics, *e.g.* SSIM and Bhattacharya coefficient [7], SSIM and PSNR [44], may strengthen the objective evaluation by accounting for the quality of both the result and the color transfer. However, figure 8.1 shows that independent objective metrics may be weak predictors of the color transfer quality. To build a stronger prediction model, we consider an ensemble of features which account for subjective aspects of the color transfer evaluation.



**Figure 8.1** – Results from two color transfer methods: (a) Pitié et al.’s [34] and (b) Hristova et al.’s [7] (our style-aware color transfer). SSIM and Bhattacharya coefficient, computed for both results, are inconsistent with our subjective evaluation.

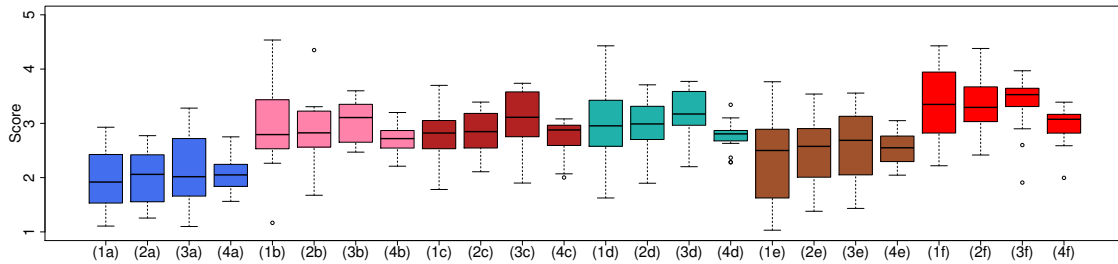


**Figure 8.2** – Main flowchart.

Flowchart of our method. A regression model, allowing to predict the quality of a color transfer, is built from subjective scores and a set of image features.

### 3 Our method

We propose a model for predicting the human judgment on results from color transfer methods. The flowchart of our method is illustrated in figure 8.2. Given triples of images (input and target images and a result), we conduct a user study and obtain subjective scores for each triple. Then, for each triple, we extract several image features. We apply a regression method between the image features and the subjective scores. That way, we build a model, predicting the color transfer quality of any image triple, given its image features.



**Figure 8.3** – Box-and-Whisker plots for: (1) the scores from our user study, (2) the predictions from regression with random forests, (3) the predictions from non-linear regression and (4) the predictions from linear regression. Each group of four identically-colored box plots represents one of six color transfer methods: (a)Pouli et al.’s [39], (b)Pitie et al.’s [42], (c)Reinhard et al.’s [3], (d)Pitié et al.’s [34], (e)Bonneel et al.’s [6], (f)Hristova et al.’s [7].

### 3.1 User study.

**Protocol.** We conducted a subjective evaluation of results from 6 color transfer methods, *i.e.* two non-parametric methods [39, 42], two global [3, 34] and two local [6, 7] parametric methods. We computed 20 results from each method, for a total of 120 results. The protocol of the user study is described hereafter. It is similar to the protocol of the user evaluation, which we conducted to assess the performance of our style-aware color transfer (see also chapter 5).

The participants were first shown tuples of input and target images. They had 5 seconds (prior to displaying the result) to get familiar with the images and to imagine what result they would expect to see. The evaluation of the results was guided by two criteria, encompassing the main aspects of the human judgment on a color transfer [7]. We were interested, first, in the way users perceived the match in color between the result and the target image and second, in users' judgment on the aesthetic quality of the result. Based on these two criteria, users were asked to give a single score reflecting their expectation about the result. Five-point scale (5-excellent, 4-good, 3-acceptable, 2-poor, 1-bad) was used in the evaluation. To avoid any possible bias, we used two repetitions per result.

Furthermore, similarly to our user study, presented in chapter 5, an extra triple, called a baseline, was inserted among the 120 results and shown randomly to each participant. The baseline was the only triple, for which the colors of the result and the target image differed significantly one from another. Therefore, users were expected to give the baseline the lowest score. That way, the evaluation of the baseline identified how trustworthy a user's judgment was.

The number of participants in the user study was 20 and the majority of them had average image editing expertise. Each user evaluated the results individually using an online platform. The order of displaying the results was random and different for each participant. A short training session took place before the real test in order for the users to get familiar with the presented task and the platform.

**Data.** For each image triple, we computed a final subjective score as the weighted mean over the scores of all participants. We used weights, which were inversely proportional to the baseline score, given by a participant. Figure 8.3 shows the score distribution for each of the six color transfer methods. According to the conducted user study, our style-aware method [7] outperforms the others by obtaining the highest mean score. Furthermore, we observe differences in the score distribution respectively between the two non-parametric methods and the two local parametric methods. In contrast, the performance of the two global parametric methods is similar.

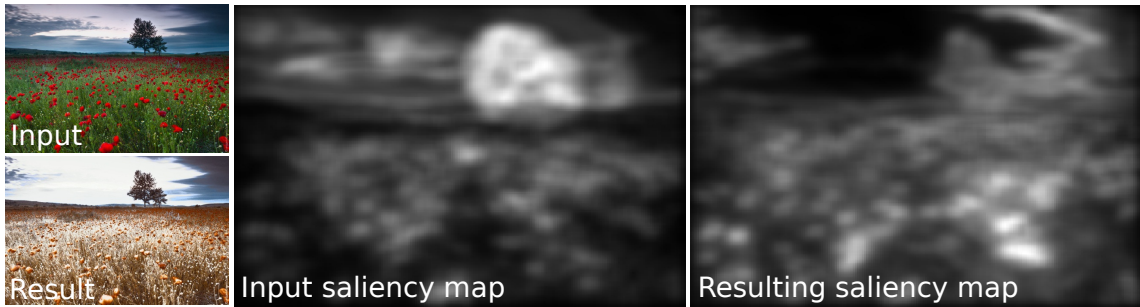
---

### 3.2 The regression model

**Objective features.** We extract nine image features for each image triple ( $\mathbf{I}$ ,  $\mathbf{J}$ ,  $\mathbf{R}$ ) in our image dataset, where  $\mathbf{I}$  denotes the input image,  $\mathbf{J}$  the target image and  $\mathbf{R}$  the result. We denote the channels of an image  $\mathbf{H}$  in the CIE Lab color space by  $\mathbf{H}_L$ ,  $\mathbf{H}_a$  and  $\mathbf{H}_b$  ( $\mathbf{H}$  stands for either of the images  $\mathbf{I}$ ,  $\mathbf{J}$  and  $\mathbf{R}$ ). The Bhattacharya coefficient between two image channels  $\mathbf{C}_1$  and  $\mathbf{C}_2$  is denoted by  $BC(\mathbf{C}_1, \mathbf{C}_2)$ . The features, considered in our model, are defined below:

- $SSIM(\mathbf{R}, \mathbf{I})$  measures the degree of artifacts in the result, caused by the color transfer. The SSIM is computed as in [7], *i.e.* it is computed on the luminance channel  $\mathbf{H}_L$  using only the contrast and structural components of the similarity metric (see also chapter 5);
- Luminance histogram similarity, computed between the images  $\mathbf{R}$  and  $\mathbf{J}$  as  $BC(\mathbf{R}_L, \mathbf{J}_L)$ ;
- Color histogram similarity, computed as the mean of  $BC(\mathbf{R}_a, \mathbf{J}_a)$  and  $BC(\mathbf{R}_b, \mathbf{J}_b)$ ;
- Saliency map similarity between the images  $\mathbf{R}$  and  $\mathbf{J}$ , computing the similarity between two saliency maps, as presented in [99]. For the sake of robustness, discussed in [100], the saliency map of each image is computed as the mean of two saliency maps, obtained with the methods in [101] and [102];
- Histogram similarity of color appearance attributes, computed as  $BC(\mathbf{R}_l, \mathbf{J}_l)$ , where  $l$  denotes one of five color appearance attributes, *i.e.* brightness, lightness, chroma, colorfulness and saturation [103].

The first three metrics represent the basic objective evaluation, previously used to evaluate our style-aware color transfer method [7] (chapter 5). They identify how the degree of artifacts in the result and the match in color and light between the result and the target image influence users' perception. Comparing saliency maps allows to test the influence of a color transfer on image saliency. In the best case, the saliency maps of the input image and the result should be the same. Figure 8.4 illustrates the impact of the color transfer on the salient areas of the result. Color transfer methods may compromise the preservation of the input saliency, as shown in figure 8.4. Therefore, saliency is one of the important indicators of the quality of the color transfer.



**Figure 8.4** – The saliency maps of an input image and a result of the color transfer method in [6]. The difference between the two saliency maps is due to the color transfer.

The channels of the CIE Lab color space represent well the light and color distributions of an image. However, they do not account for color appearance phenomena [103], *e.g.* chromatic adaptation, Hunt effect, Stevens effect, etc., which occur with a change in the viewing conditions. In contrast, the color appearance attributes describe the perceptual aspects of color. The nine objective features represent the set of the independent variables in our model.

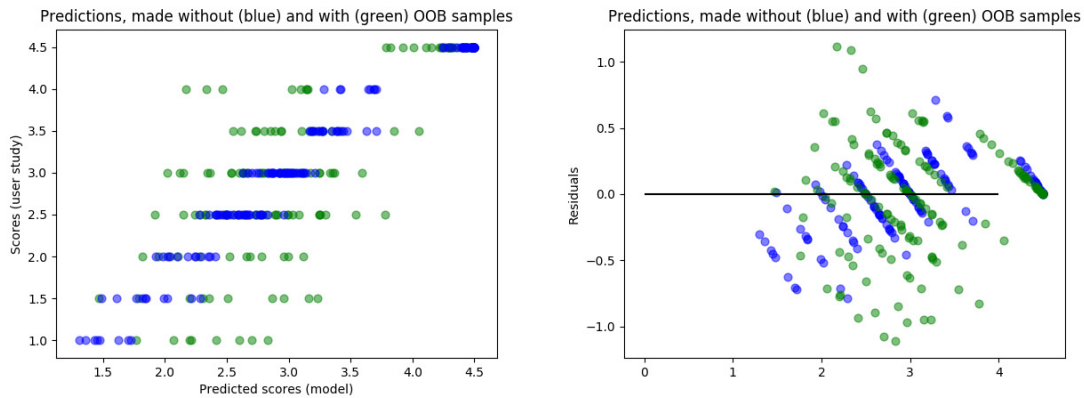
To apply a regression method, we first carry out a proper preprocessing of the data, collected from our user study, as presented hereafter.

**Quantization and over-sampling.** Due to the use of baseline weights, the set of subjective scores is highly under-sampled as there is an insufficient number of triples per score. To tackle this issue, we round the scores up to the nearest 0.5 so that at least two triples have then the same score. Despite the performed quantization, the set of subjec-

tive scores remains imbalanced. As discussed in [104], imbalanced data may compromise the performance of regression methods, such as discriminant analysis and decision trees. Therefore, we balance our data by using the synthetic minority over-sampling technique [105]. In our model, the balanced set of scores represents the set of dependent variables.

**Regression methods.** To fit our data by a simple linear model, we first use linear regression. Our experiments with linear regression, however, have pointed out strong non-linearities in the set of subjective scores and image features. That is why, we also apply non-linear regression. We use support vector regression with radial basis function kernel [106].

To further improve the accuracy of the fit, we use random forests [107]. Random forests is a learning method, which constructs an ensemble of decision trees. Although decision trees provide a classical model for fitting various data, they tend to over-fit the training sets and introduce a high variance. Random forests aim to correct the high variance of decision trees and provide a more accurate prediction by using bootstrapping. Random forests take into account strong non-linearities in the data and are not sensitive to correlated predictors.

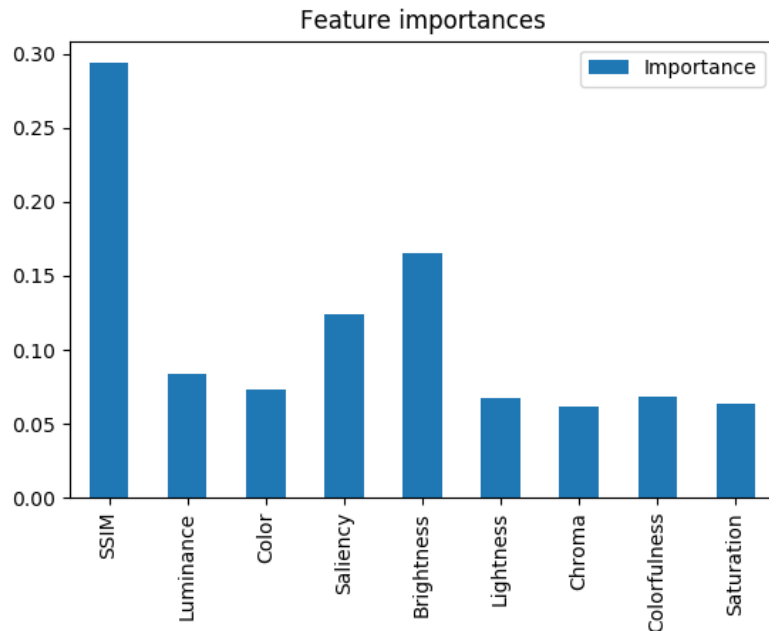


**Figure 8.5** – Left plot: relationship between the predicted scores from our model and the actual scores from our user study. Right plot: residual distribution in our model.

## 4 Results

**Cross-validation.** To analyze the behavior of each of the three regression methods, we performed k-fold cross-validation with 10 splits. For each regression method, table 8.1 shows the mean correlation between predicted scores and actual subjective scores, and the mean square error (MSE), computed over all test sets. Linear regression is the least correlated with our data method. The regression with random forests provides the most correlated with the subjective scores prediction.

The same conclusion is drawn from the box plots in figure 8.3. The predicted score distributions from the linear regression is characterized by a low variance and significantly differs from the distribution of subjective scores per color transfer method. Overall, the prediction, made with non-linear regression, is much more accurate than the prediction,



**Figure 8.6** – Bar plots of the importance of each objective feature in our model.

	Random forests	Linear	Non-linear
<b>Corr</b>	$0.765 \pm 0.133$	$0.567 \pm 0.135$	$0.644 \pm 0.157$
<b>MSE</b>	0.472	0.808	0.889

**Table 8.1** – Correlation  $\pm$  standard deviation and MSE over all 10 test sets in our cross-validation.

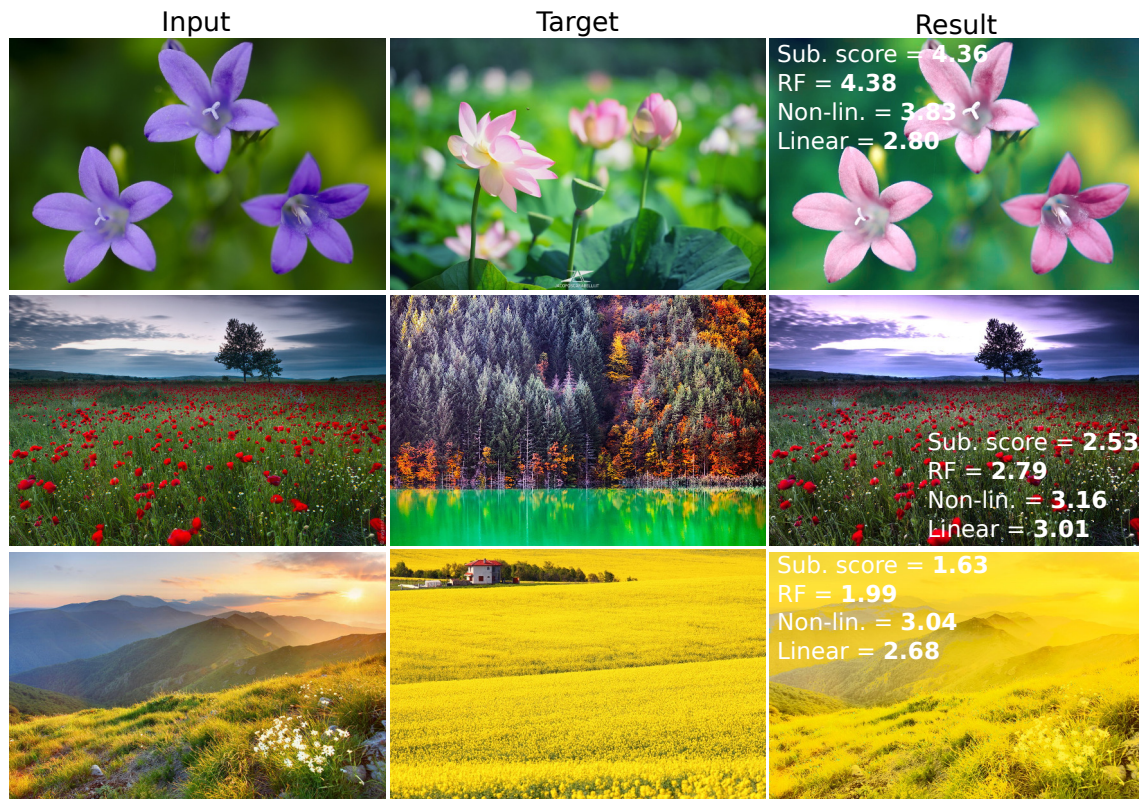
made with linear regression. However, non-linear regression fails for our style-aware local method [7]. The most consistent regression method is the random forests. It provides the best fit for the score distributions of all six color transfer methods.

Furthermore, figure 8.7 illustrates the actual and predicted scores for three representative triplets of images from our user study. Random forests provide a very accurate score, close to the actual subjective score. For most image triplets, the linear regression gives the poorest prediction, followed by the non-linear regression.

Our analysis has shown that regression with random forests describes very precisely the relationship between the users' evaluation and the nine objective image features and therefore, we adopt it in our model. Hereafter, we discuss in more details the accuracy of this regression model.

**Random forests accuracy.** The accuracy is characterized by the coefficient of determination equal to 0.94, the out-of-bag (OOB) error equal to 0.58, and the mean square error (MSE) of the prediction equal to 0.063. The high coefficient of determination indicates that a major part of the variability of the subjective scores is explained by our model. The left plot in figure 8.5 illustrates a strong correlation between the predictions (made with and without using OOB samples) and the actual user scores. The right plot in figure 8.5 shows the residuals of the predictions (made with and without using OOB samples). The residuals are symmetrically distributed around the x-axis and are clustered around -1.5 and 1.5 on the y-axis. The random patterns in the residual plot indicate that





**Figure 8.7** – Score predictions from random forests (RF), linear and non-linear regressions (Non-lin.). Color transfer method used (from first to third row): [7], [3], [6].

our model provides a decent fit of the data.

**Feature importances.** Despite the lack of an analytical model in the random forests, a summary of the importance of each predictor can be computed [107]. The plot in figure 8.6 shows the importance of each image feature in our dataset. SSIM has a significantly greater importance than the second most important feature, the brightness histogram similarity. This shows that users are unlikely to give high scores to a color transfer, which compromises the integrity of the result. The first three most important features, *i.e.* SSIM, brightness histogram similarity and saliency map similarity, are all perceptual features. Surprisingly, the color histogram similarity plays a much less significant role in our model. This is due to the fact that the Bhattacharya coefficient represents the overlap between color distributions in the CIE Lab color space without accounting for various color appearance effects.

## 5 Conclusion

In this chapter, we have presented a regression model, based on random forests, for objective evaluation of results from color transfer methods. Our method describes the relationship between the scores from our user study and nine image features, used for objective image evaluation. Our analysis has shown that perceptual image features, such as SSIM and saliency, play a main role in predicting the color transfer quality. Furthermore, a cross-validation indicates a high correlation between predicted and actual scores. To this

end, our regression model can be used as a general prediction of a user’s judgment on any color transfer result.

The proposed metric accurately assesses the quality of color transfer methods. However, our regression model would fail to predict the quality of style transfer methods, based on deep learning [52, 56]. This is due to the fact that our model does not use multi-layer image feature information. To improve the proposed perceptual metric, in the future we could tackle high-end features, such as image aesthetics and image composition. Furthermore, to adapt the metrics to example-guided texture transfer methods [55], texture descriptors could be added in our regression model.





## **Part III**

### **Example-guided high-dynamic-range image editing**



# Introduction

All color transfer methods, presented in the previous part (including proposed and state-of-the-art methods), are applied to low-dynamic-range (LDR) images, *i.e.* images whose luminance values lie inside the displayable range. In this part, we extend the presented color transfer methods to high-dynamic-range (HDR) images and introduce an example-guided method for computing HDR images from two LDR images.

HDR images capture the luminance of real-world scenes, which ranges from extreme dark to direct sunlight. Details of both shadow and highlight areas, present in a high-dynamic scene, can be recovered in a single HDR image. LDR devices are unable to display the luminance range of real-world scenes. To this end, tone mapping operators are applied to map the large luminance variations of the HDR images to the displayable range. However, this type of compression results in a loss of details and may cause the appearance of structural artifacts. As the plausibility of the tone mapped image cannot be guaranteed, any modifications to the color and light of the HDR images, *e.g.* color transfer, need to be handled directly in the HDR domain.

In chapter 9, we extend state-of-the-art color transfer methods to the HDR domain. First, we present an HDR extension to our style transfer method [7], which was introduced in chapter 5, part II. Our style transfer method consists of steps which are incorporated (independently or as a combination) in the frameworks of other color transfer methods. Therefore, the HDR extension, proposed in chapter 9, can be easily adapted to a number of existing color transfer methods. The experiments, shown in the end of chapter 9, indicate that the proposed HDR extension performs better than the direct application of the color transfer methods to HDR images.

HDR images are commonly created using standard digital imaging. Standard digital imaging produces LDR images in which the luminance dynamics is replaced by the discrete luma range. The latter limits the capture of details in scene shadow and highlight regions, resulting in an under-/over-exposure. The camera response function (CRF) gives the relationship between the luminance and the luma up to a scale factor. To recover an HDR image using standard digital imaging, we need 1) to estimate the CRF and 2) to recover the details in black/white image pixels. The most common way of creating an HDR image by respecting these two requirements is to merge multiple LDR images, taken at various exposure times and referred to, in this part, as multi-exposure images.

Despite the efficiency of multi-exposure methods, the process of creating an HDR image is usually time-consuming. First, users are required to use a tripod and to adjust the camera exposure time each time they take an image (if the camera does not include exposure bracketing function). Moreover, during the shooting process, misalignment could become an issue, especially when there are moving objects in the scene. To this end, more time is likely to be spent aligning the images in the hope of correcting ghosting artifacts.

To represent the atmosphere and the details of a real-world environment, users often

need to take more than three exposure images [108]. However, this may increase the risk of misalignment and noise. In the particular case of dark environments with a high luminance range, decreasing the exposure time allows to capture fine details in the highlights, but may significantly increase the levels of noise in the computed HDR image.

To overcome the main limitations of the multi-exposure approach, in chapter 10 we propose a method for automatic creation of HDR images from only two LDR images - a non-flash image and a flash image. Using an approximation of the CRF, we first compute a sequence of multiple images at various brightness levels from the non-flash image. Then, we recover the details in the under-/over- exposed pixels of these images using the flash image embedded in our novel CAT method, called bi-local CAT. The final sequence of multi-exposure images is merged together to recover an HDR image. Our method is mainly applied to dark scenes with high-dynamics for which the reach of the flash is significant. As presented in chapter 10, we recover the dynamic range of HDR images and obtain noise-free HDR images with high quality. Apart from HDR image creation, our method can be also employed in the context of example-guided image enhancement.

# Color transfer between high-dynamic-range images

# 9

---

## 1 Introduction

This chapter introduces a transfer of color and light between two HDR images. As outlined in chapters 5, 6 and 7 in part II, a number of color transfer methods provide various solutions to example-based transfer of color, light and gradient between pairs of LDR images. However, the direct application of these methods to the HDR domain is not evident. In this chapter, we extend the LDR color transfer methods to the HDR domain by introducing series of adaptation techniques. The chapter focuses mainly on the extension of our style-aware color transfer method [7]. As presented in chapter 5, our style-aware method is composed of two main steps, *i.e.* a color transformation and a local CAT. The two steps are incorporated (independently or as a combination) in the framework of other color transfer methods. To this end, the adaptation techniques, developed in this chapter, serve as a basis for extending a wider class of color transfer methods to the HDR domain. Furthermore, experiments show that results obtained with the proposed HDR extension exhibit less artifacts and are visually more pleasing than results obtained by a straightforward application of the color transfer methods.

The chapter is organized as follows. Section 2 discusses the main drawbacks of applying state-of-the-art color transfer methods to HDR images. Extensions to the HDR domain of state-of-the-art color transfer methods are proposed in section 3, followed by results and evaluation. Finally, section 4 concludes the chapter.

**Publication.** The work, presented in this chapter, has been published in the following paper:

*H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, "Color transfer between high-dynamic-range images." SPIE Optical Engineering Applications. International Society for Optics and Photonics, 2015.*

---

## 2 Why do LDR color transfer methods need to be extended to HDR images?

The state-of-the-art color transfer methods have been so far limited to LDR images and they cannot be directly applied to the HDR domain due to several restrictions, as discussed hereafter.

The color transfer between LDR images is carried out in various color spaces. For instance, Reinhard et al. [3] have designed  $l\alpha\beta$  to decorrelate light and colors of natural

LDR images. The color space  $l\alpha\beta$  is derived as a linear transformation of CIE XYZ. The color space CIE XYZ uses imaginary primaries such that values in a low-dynamic-range interval (usually  $[0, 1]$ ) cover the visible gamut. Therefore, values much greater than the upper bound of this low-dynamic-range interval have to be accommodated as to include the desired high dynamics of the HDR images.

The color space CIE Lab is widely used in a number of recent color transfer methods [6, 7]. Reinhard et al. [109] have conducted experiments showing that CIE Lab is the best color space for carrying out a color transfer. However, CIE Lab is limited to color stimuli with luminance levels from zero to perfect diffuse white. This means that CIE Lab predicts the color trend for luminance levels below and around the display white point [10]. Therefore, the applicability of CIE Lab to HDR images is uncertain.

Furthermore, as discussed in chapter 5, global color transfer methods fit the light and color distributions using the MGD and assume that the MGD parametric model can account for the luminance variations in the entire image. However, a unique MGD can hardly fit the large luminance range of the HDR images and be representative of it. As elaborated in chapter 5, the multivariate Gaussian assumption may not always hold in the LDR domain. To handle this issue, local LDR color transfer methods cluster images into Gaussian clusters [5–7]. The image luminance is often used to carry out the image clustering and the mapping between the clusters (as it is the case of our style-aware method [7]). Thanks to the low dynamics of the LDR images, luminance is often approximated with lightness and vice versa. Nevertheless, in the context of HDR imagery, absolute luminance (measured in  $cd/m^2$ ), relative luminance (relative to the perfect diffuse white) and lightness need to be distinguished and accommodated properly in the frameworks of the color transfer methods. To extend these methods to the HDR domain, new clustering and mapping strategies should be developed.

The following section presents an extension to HDR images of our local color transfer method [7], presented in chapter 5. Independently or as a combination, the steps of our method are integrated in the frameworks of state-of-the-art color transfer methods. Therefore, the adaptation of our color transfer method to HDR images guides the adaptation for a number of other color transfer methods.

---

## 3 Adapting a color transfer method to HDR images

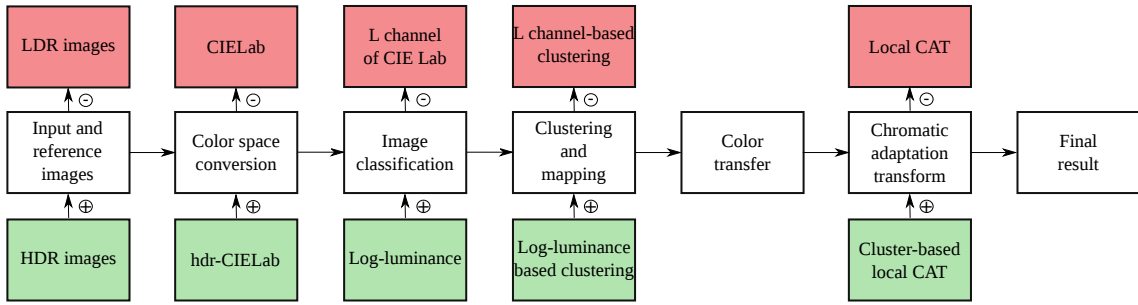
In the present section, we discuss the steps of our style-aware color transfer method and focus on their extension to HDR images by taking into account the aforementioned drawbacks.

---

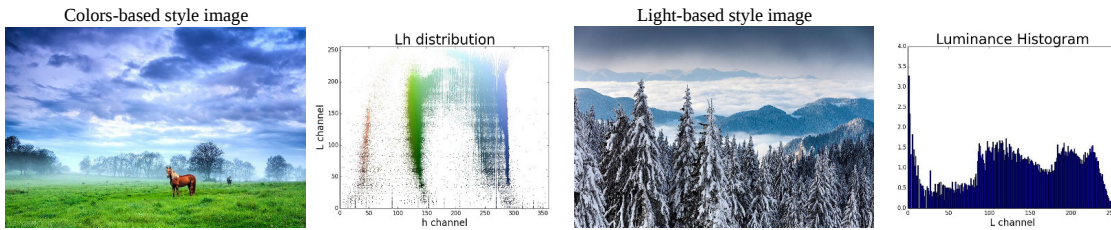
### 3.1 Style-aware color transfer [7]

First, we recall the steps of our local style-aware method [7] in figure 9.1. Before performing any color transformations, the method classifies and clusters both input and target images. Our image classification algorithm detects the main features (among the two considered - light and colors) of the input and target images. Our method classifies the images into one of two types: light-based style images and colors-based style images, as illustrated in figure 9.2. The classification is carried out in CIE Lch color space and aims





**Figure 9.1** – The framework of our style-aware color transfer method [7] (the steps of the method are displayed in the middle boxes). The red boxes beneath each step illustrate the parts of our original method replaced by modifications in our extended HDR method. The modifications are shown in the green boxes.



**Figure 9.2** – A colors-based style image and its Luminance-Hue distribution with well-defined color clusters (on the left). A light-based style image and its luminance histogram, identifying the presence of shadows, midtones and highlights in the image (on the right).

to find the significant peaks in the hue histogram of the set of non-gray image pixels.

Once both input and target images are classified, they are separated into Gaussian clusters using Gaussian mixture models. The clustering is performed either on the (1D) luminance histogram or on the (2D) Luminance-Hue joint distribution (depending on the image type). In our style-aware method, we apply one of four designed mapping policies to properly map the obtained input and target clusters. Similarly to Bonneel et al. [6], we apply luminance-based mapping to map the clusters of two light-based style images. The other three mapping policies jointly consider the luminance and the hue of the input and target images to reflect the key ideas behind commonly used photographic techniques.

As in [6], we apply Pitié’s color transformation [60] on the  $a$  and  $b$  channels of CIE Lab. We apply our local CAT as a final step of the method to preserve the photo-realism of the result and to reduce the transfer of false colors. Local CAT is performed pixel-wise on the pixels of the input image. The target white point is computed by assuming Gray World [18] (Gray World is discussed in part 2), whereas the input illuminant is computed in the form of a “white” image by performing Gaussian low-pass filter. For more details regarding the steps of our method, please refer to chapter 5.

### 3.2 Extension to the HDR domain

We adapted our style-aware method to HDR images on several stages. Each stage refers to modifications of a step in the framework of the method. Figure 9.1 illustrates the proposed modifications for extension of our local style-aware method to HDR images.

### 3.2.1 Color space conversion

Our style-aware method is carried out in CIE Lab. The color space offers a good representation of the lightness and the chroma of the LDR images. However, for luminance values far beyond the perfect diffuse white, CIE Lab is no longer able to reproduce well the image color gamut. To address this issue, we follow the recommendations of Fairchild et al. [110] and we replace the cubic root function of the  $L$  channel of CIE Lab with the Michaelis-Menten function [110] (denoted as  $f(y)$ ) as follows:

$$f(y) = 247 \frac{y^\epsilon}{y^\epsilon + 2^\epsilon} + 0.02 \quad (9.1)$$

where  $y$  denotes the relative luminance (obtained by scaling the absolute luminance by the perfect diffuse white) ranging from 0 to 4, and  $\epsilon$  is equal to 0.58.

The chroma channels  $a$  and  $b$  are then computed by replacing the cubic root function in their standard equations (presented in part 2) by equation 9.1 and scaling them by 1/100. That is how we obtain the per-channel equations of hdr-CIELab color space [110] as an extension of CIE Lab in the HDR domain. The color space hdr-CIELab is specially developed to predict the color trend above the diffuse white for images with high dynamic range.

### 3.2.2 Image classification

Once a good prediction of the HDR color gamut is ensured, we carry out the classification step of our method [7]. Apart from detecting the main features of images, the classification algorithm determines the number of clusters which are passed to the clustering step. For colors-based style images, the number of color clusters is determined using their hue histograms. In the HDR image extension, we compute the hue as a transformation of the  $a$  and  $b$  channels of the hdr-CIELab color space. That way, the classification algorithm will properly determine the number of significant color clusters in the HDR images, and thus, it will properly detect the main feature of a given image.

Regarding the number of luminance clusters in light-based style images, they correspond to the number of significant peaks in the luminance histogram. Instead of using the luminance histogram, in the extension of the method to HDR images we adopt the log-luminance histogram. We justify this choice in the following section.

### 3.2.3 Clustering and mapping

The clustering step of our style transfer method partitions both the input and target images into Gaussian clusters using GMMs. The clustering is performed using either the luminance histogram or the Luminance-Hue joint distribution. As in the LDR domain absolute luminance and relative lightness are practically interchangeable, the  $L$  channel of CIE Lab color space (*i.e.* the lightness) is often used to cluster the input and target images [6, 7]. However, in the context of HDR imagery, the  $L$  channel of hdr-CIELab refers to the lightness, *i.e.* the relative luminance, and not the absolute luminance. Therefore, instead of adopting the  $L$  channel (equation 9.1) for carrying out the image clustering in the HDR domain, we use the luminance and more precisely, the logarithmic transformation of the absolute luminance.

As illustrated in equation 9.1, the  $L$  channel of  $\text{hdr-CIELab}$  is a function of the relative luminance  $y$ , which is obtained by scaling the absolute luminance (in  $\text{cd/m}^2$ ) by the diffuse white. Moreover, the logarithmic function of the absolute luminance is a good approximation of the brightness [111]. As a monotonic transformation, the logarithmic transformation preserves the locations of the minima and the maxima in the histogram of the absolute luminance. To this end, we can use the logarithmic approximation of the brightness in the place of the  $L$  channel of the  $\text{hdr-Lab}$  color space in both clustering approaches (for both light-based and colors-based style images).

The log-luminance histogram is also adopted for the purposes of finding the number of luminance clusters during the classification process. Moreover, it is then used to map the target to the input clusters after the image clustering.

Once we carry out all the described adaptation operations, we perform the color transformation between each pair of corresponding clusters, as described in section 3.4. The transformation is a closed-form solution to Monge-Kantorovich optimization problem. The transformation between the input and target images is linear and therefore, no additional modifications are required regarding the color transformation.

Finally, in the last step of our style-aware method, we apply a local CAT, which is replaced by a new cluster-based CAT in the HDR extension of the method. The cluster-based local CAT can be either integrated in a color transfer framework or applied as a standalone color grading technique. The new CAT method is detailed in the following section.

### 3.2.4 Cluster-based local CAT

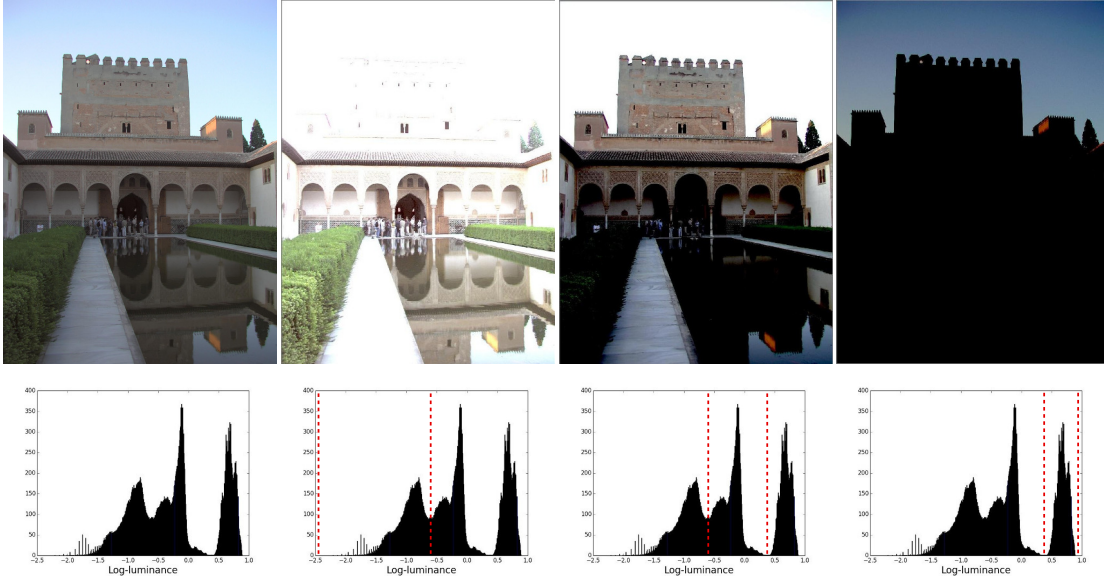
CAT algorithm adapts the colors of a given image to a target illuminant, which usually is one of well-known white points: D65, D50, etc. More details are presented in part 2.

To adapt the colors of an HDR image to a well-known illuminant, we can compute an estimation of the target white point. However, due to the high luminance range of the HDR images, one global white point may not represent well enough the luminance variations of the scene. To tackle this issue, we propose a new cluster-based local CAT. We propose a partitioning into regions according to the HDR luminance distribution. Then, from the luminance values of each region, we compute local target white points.

The number of regions resulting from the partitioning of an HDR image depends on its luminance range. We compute the luminance histogram of a given image in the log-domain (the choice of the log-domain is justified in section 3.2). The number of peaks in the log-luminance histogram corresponds to the number of differently illuminated regions in an image (*e.g.* highlights, shadows, etc.) as illustrated in figure 9.3.

An HDR image is partitioned in the log-domain using its log-luminance histogram. Two parameters are considered for finding the peaks of the log-luminance histogram: the minimum histogram peak value  $s_{min}$  and the minimum distance between two peaks  $d_{min}$ . The number of bins in the log-luminance histogram is set to  $32 \times r$  ( $r$  is the range of the log-luminance histogram [112]). Furthermore, we follow the recommendations of Boitard et al. [112] and set the parameters  $s_{min}$  and  $d_{min}$  respectively to  $\frac{nm}{16r}$  (where  $n$  and  $m$  are the two image dimensions) and 0.65.

Once the minima between each two peaks are defined, they are set as region limits. Each image region consists of pixels with three coordinates  $(X, Y, Z)$ , for which  $\log(Y)$



**Figure 9.3** – An HDR image and its three regions, corresponding to the peaks of its log-luminance histogram (we use Reinhard et al.’s tone mapping operator [111] to display the HDR image and its regions). Two red dashed lines define the region limits in the log-luminance histogram.

lies within the limits of the region. To make a smooth transition between the different regions after the chromatic adaptation, overlapping between them is performed. All of the pixels with log-luminance values within a small offset  $\delta$  from a given limit, are considered as overlapping pixels. The value  $\delta$  is given in the log-domain and it is set to 1. Each overlapping pixel is assigned two weights, measuring the belonging of the pixel to each of two image regions. The weight of an overlapping pixel for a given region is derived as follows [112]:

$$\omega = e^{-\frac{(\log(Y) - \log(l))^2}{2\sigma^2}}, \sigma = \frac{\delta}{2\sqrt{2\log(3)}} \quad (9.2)$$

where  $Y$  is the luminance of the pixel and  $l$  stands for the limit of the given region.

Moreover, the white point of each region in the target HDR image is computed as follows [112]:

$$X_{Wr} = e^{\frac{1}{s_j} \sum_{i=1}^{s_j} \log(X_r + \delta_1)}, Y_{Wr} = e^{\frac{1}{s_j} \sum_{i=1}^{s_j} \log(Y_r + \delta_1)}, Z_{Wr} = e^{\frac{1}{s_j} \sum_{i=1}^{s_j} \log(Z_r + \delta_1)} \quad (9.3)$$

where  $s_j$  is the size of region  $j$  and  $\delta_1$  is a small offset.

Once the target white points are computed, they are mapped to the input regions. We define the center value of an input region as the maximum  $\log(Y)$  value within the limits of the region. The target white point with the closest  $\log(Y)$  value to the center of an input region, is mapped to that input region.

The iCAM CAT, introduced in part 2, is performed locally using an estimation of the input white point for each pixel in the input image. Unlike Kuang et al. [21], the input white point, called white image, is computed using a low-pass Gaussian filter with a kernel size equal to the sum of the two input image dimensions.



**Figure 9.4** – Results from applying our cluster-based CAT algorithm. For this pair of input/target images, CAT algorithm converges in 20 iterations. After the first 7 steps, the input image is already adapted to the target illuminant. However, there is still a cast of the greenish input illuminant which is removed by the time the algorithm converges.

To sum up, the proposed CAT algorithm is performed locally for each pixel of the input image using the closest (to the input region) local target white point (equation 9.3). The process is iterative and it is repeated until convergence. The convergence criterion is given as follows:

$$\Delta = \frac{\|I_W^k - I_W^{k-1}\|_F}{\|I_W^{k-1}\|_F} \quad (9.4)$$

where  $\|\cdot\|_F$  is the Frobenius norm,  $I_W$  refers to the input white image, and  $k$  is the number of the current iteration. The iterative process stops when  $\Delta$  gets lower than  $10^{-3}$ . Usually, the number of iterations does not exceed 20. Figure 9.4 illustrates how the result of applying cluster-based local CAT changes with the increase of the iterations.

### 3.3 Results

Several experiments have been conducted to test the efficiency of the proposed extension as discussed hereafter.

#### 3.3.1 Protocol

We apply independently the color transformation and the local CAT algorithm to show their impact on the color transfer (for both our style transfer method and its extension). Then, we carry out all the steps of our HDR extension method (color transformation plus local CAT) and compare the obtained results with results, obtained from a directly applying our style transfer method to HDR images. Results are shown in figure 9.5.

Furthermore, we use a set of ten image pairs to obtain 10 HDR image results for both our local LDR method (directly applied to HDR images) and its HDR extension. To evaluate the efficiency of the proposed extension, an objective evaluation has been carried out. The evaluation is performed on the ten image pairs for both the extended and the original methods as explained in the following section.

#### 3.3.2 Objective evaluation

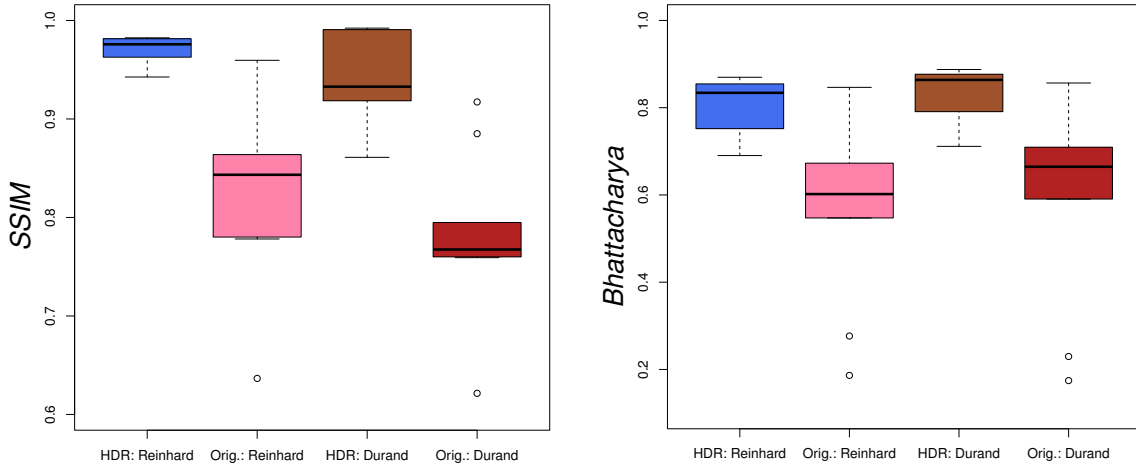
There exists a perceptual metrics, called HDR-VDP-2 [113], for comparing two HDR images. HDR-VDP-2 indicates the difference in the perceived luminance of both HDR images and it is not reliable when it comes to computing the color similarity between the





**Figure 9.5** – Results obtained by applying our style-aware color transfer method [7] (without any modifications) and its extension to HDR images. From left to right: results of applying the color transformation without the local CAT as a final step; results of applying local CAT in iterative manner (without color transformation); result of applying both color transformation and local CAT. All images are displayed using the tone mapping operator by Reinhard et al. [111].

two HDR images. Moreover, HDR-VDP-2 can only be used to compare the luminance of the result with the luminance of the input image. However, we are mostly interested in comparing the obtained results with the target images so that we can evaluate the quality of the proposed color transfer. Therefore, we evaluate the match in the color palettes between the results and the target images in the LDR domain after tone mapping. To evaluate how successful a color transfer is and whether it introduces structural artifacts to the final result, we use two complimentary metrics, Bhattacharya coefficient [69] and SSIM [68] respectively (we used a combination of these metrics for the objective evaluation, presented in chapter 5). On one hand, we apply SSIM on the tone-mapped result and the tone-mapped input image to measure the degree of artifacts in the final result. On the other hand, we apply Bhattacharya coefficient to evaluate how close the color and light distributions of the tone-mapped result are to those of the tone-mapped target image. As the displayed result strongly depends on the tone mapping operator, two tone map-



**Figure 9.6** – Box-and-Whisker plots of both SSIM and Bhattacharya coefficient for results obtained using our style-aware color transfer method (without modifications) and its HDR extension.

ping operators, Reinhard et al.’s [111] and Durand et al.’s [8], are used in the evaluation. Figure 9.6 presents the Box-and-Whisker plots of the SSIM and Bhattacharya coefficient marginal distributions for each of the two methods (the original method and its HDR extension) and for each of the two tone mapping operators.

The results obtained with the HDR extension preserve the structure of the input image. In contrast, if we apply the original LDR method directly to HDR images, the degree of artifacts caused by the color transfer (measured by the SSIM), increases. This analysis holds for both tone-mapping operators [8, 111] and moreover, it is supported by the results in figure 9.5. The loss of structural details as well as the presence of artifacts (both caused by the direct application of our style-aware method to HDR images) are visible in images (d) and (f). Even more, they are clearly noticeable on an HDR display as well. In contrast, the proposed HDR extension succeeds in preserving the structural details of the input image, as results (a) and (b) show.

The results obtained with the HDR extension have significantly higher Bhattacharya coefficients than the results obtained from directly applying our style transfer method to HDR images (for both tone mapping operators). In comparison to the colors of image (d) in figure 9.5, the colors of image (a) are much closer to the target color palette. Finally, the cluster-based local CAT helps to better represent the target illuminant, which results in a more accurate transfer of the target color and light (figure 9.5).

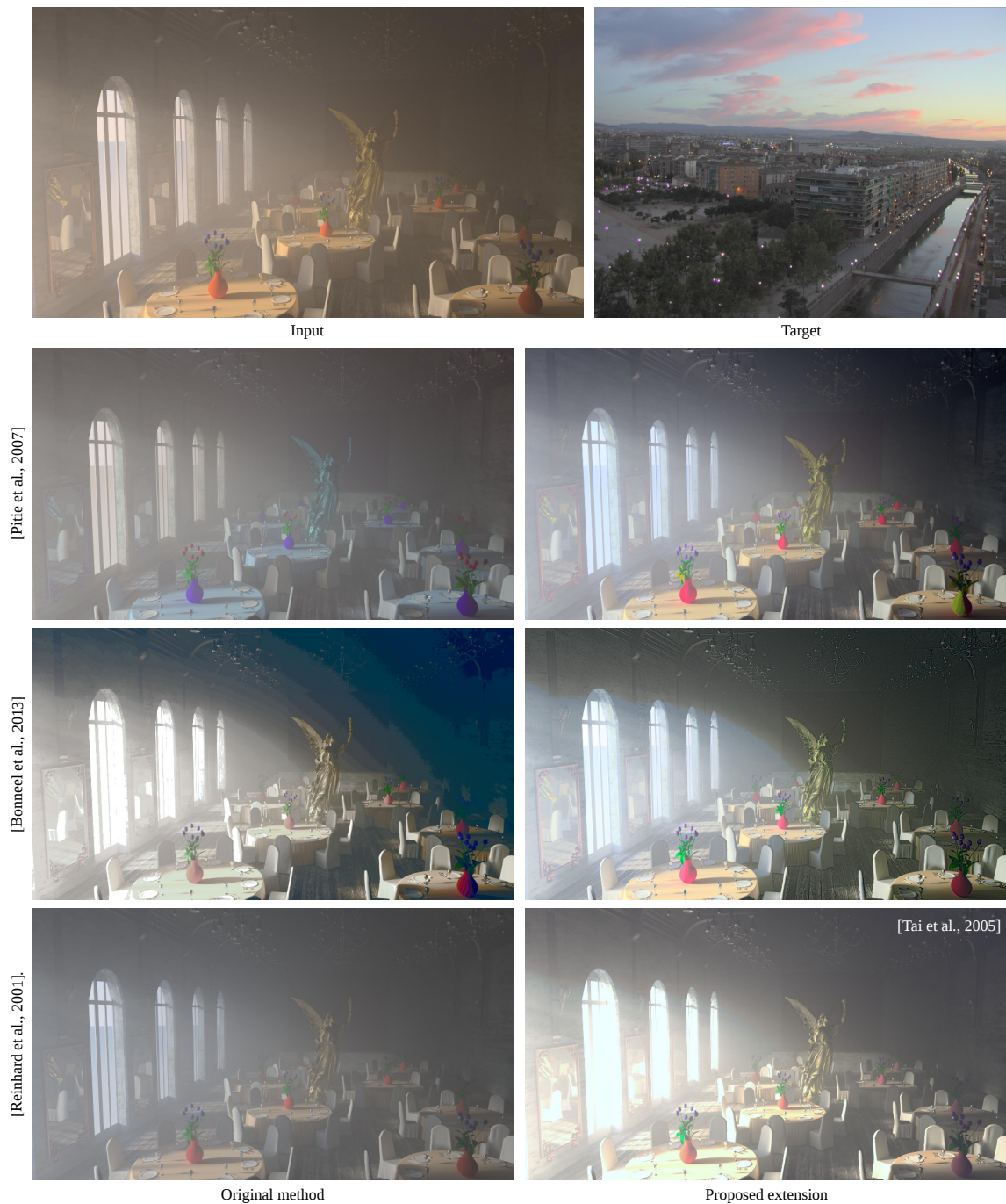
### 3.4 Is the proposed extension applicable to state-of-the-art color transfer methods?

In the previous section, we detailed and analyzed the extension of our local style transfer method [7] to the HDR domain. The proposed modifications serve as a basis for extending state-of-the-art color transfer methods to HDR images. Depending on the used method, we recommend different types of modifications, as shown in table 9.1.



Method	LDR domain	HDR domain
Reinhard et al. [3]	Global method	Local method (Tai et al. [5])
Pitié et al. [60]	CIE Lab	hdr-CIELab
Bonneel et al. [6]	Luminance-based clustering	Log-luminance-based clustering
Tai et al. [5]	$l\alpha\beta$	$l\alpha\beta$ -extended

**Table 9.1** – General modifications for adapting color transfer methods to HDR images. The column named **LDR domain** displays the steps of the corresponding methods which are modified in the way shown in column **HDR domain**.



**Figure 9.7** – Results, obtained by applying state-of-the-art color transfer methods (directly, without any modifications), and results, obtained with their extensions to HDR images. Reinhard et al.'s tone mapping operator [111] is used to display the images.

First, Reinhard et al.'s transformation is built upon the assumption that the image color and light distributions can be fitted by a multivariate Gaussian distribution with a diagonal covariance matrix. This assumption does not hold in the HDR domain due to the high luminance variations of HDR images. Therefore, there is a need to use more than one parametric Gaussian model to fit the luminance distribution of the HDR images. Consequently, to enhance the effect of the color transfer, we propose to carry out Reinhard et al.'s method [3] in a cluster-based manner. We can either adopt luminance-based clustering (like in [6] and in our style transfer method for light-based style images [7]). Tai et al. [5] have already proposed a local extension of Reinhard et al.'s global method using luminance-based clustering. To this end, we extend Reinhard et al.'s method to the HDR domain using Tai et al.'s local color transfer method. In this case, an extension to  $l\alpha\beta$  color space is recommended for accommodating the high luminance range of HDR images. The extended  $l\alpha\beta$  color space should accommodate luminance values outside the displayable luminance range.

Furthermore, following the modifications, presented in section 3.2, we replace CIE Lab color space with its HDR extension (hdr-CIELab) in both Pitié et al.'s [60] and Bonneel et al.'s [6] methods. We recommend the clustering step of Bonneel et al.'s method to be carried out on the logarithmic transformation of the absolute luminance rather than on the  $L$  channel of hdr-CIELab.

Figure 9.7 shows results of a color transfer with and without the proposed modifications (using state-of-the-art color transfer methods). As hdr-CIELab predicts better than CIE Lab the color gamut of HDR images, the reference color palette is well transferred to the result for both Pitié et al.'s and Bonneel et al.'s extended methods. Furthermore, if we apply Bonneel et al.'s color grading method directly to HDR images, visible artifacts are observed. On the other hand, if we carry out the proposed modifications (regarding CIE Lab color space and the clustering step) to the former method, the degree of artifacts is lessened. This is a result of the more precise log-luminance-based clustering. Finally, as expected, Tai et al.'s method accounts for the high luminance range in HDR images and therefore, it yields more plausible results than Reinhard et al.'s method when applied to HDR images.

## 4 Conclusion

In this chapter, we have presented extensions to state-of-the-art color transfer methods to HDR images. The extensions include modifications of traditional color spaces as well as of the clustering and the mapping steps in local methods. Moreover, we have introduced a novel cluster-based chromatic adaptation transform which could be used as a standalone color grading method. Experiments have proved that when applied, the extensions of the methods yield more plausible results than the results obtained with the direct application of the LDR methods to HDR images. However, there is a room for improvements. Our experiments have shown the need to create a more precise color mapping between two HDR images. Moreover, this chapter introduced modifications to already existing color transfer methods to improve their applicability to the HDR domain. The development of a special color transformation between HDR images is an interesting direction for further improvement in the future. To this end, we consider that this chapter is an important first

step towards bridging the gap between the color transfer domain and the HDR domain.

# High-Dynamic-Range Image Recovery from Flash and Non-Flash Image Pairs

10

---

## 1 Introduction

In this chapter, we propose a method in which we use only two images to recover an HDR image - a non-flash image, taken at a certain exposure value, and its corresponding flash image. Our method can also be used for low-dynamic scenes to enhance the quality of a non-flash image with the help of a flash image. Non-flash images represent the genuine atmosphere of the original scene lighting. However, especially for images shot in dark environments, the non-flash images are often noisy and lack important details (in under-/over-exposed pixels). In contrast, flash images contain more details, but they do not preserve the original scene lighting.

The first key idea behind our HDR image creation lies in mimicking the CRF by a brightness function. The brightness function, used in our method, aims to represent the human perception of a scene at various brightness levels. This corresponds to the main purpose of digital cameras. Therefore, we assume that the CRF can be well-approximated by our brightness function. To this end, we alter the brightness of a non-flash image by a one-parameter-dependent gamma correction and yield a sequence of multiple brightness images. To create an HDR image, we eventually need to recover the missing details of the multiple brightness images (for which we recover no information by the brightness correction).

The second key idea of our method consists in recovering these missing details by using reliable information from the flash image. To retain the original ambience of the scene while preserving the details of the flash image, we propose a novel bi-local chromatic adaptation transform (CAT). The bi-local CAT is directly applied to the flash image in order to adapt its brightness to that of the non-flash image. As the non-flash brightness is lower than the flash brightness, the bi-local CAT remaps the flash pixel values into values, corresponding to lower brightness. Therefore, to allow for an increase of the brightness dynamics and the contrast of the flash image, we carry out the bi-local CAT on each of the multiple brightness images. That way, we obtain final multi-exposure images, which we merge into an HDR image. We apply our method to dark environment scenes with high dynamic range, for which the reach of the flash is significant.

The main contributions of the method, presented in this chapter, are fivefold:

- Automatic non-flash image brightness correction;

- Bi-local CAT for automatic creation of multi-exposure images from only two images - flash and non-flash;
- Automatic recovery of HDR images from the computed multi-exposure images;
- Enhancement of a non-flash image using a flash image;
- Automatic removal of the soft shadows of the flash image as well as diminution of flash reflections.

The advantages of our method over the classical multi-exposure methods are the following: 1) the number of images for obtaining an HDR image is reduced to two; 2) the usage of a tripod is not required in the case when the flash and non-flash images can be taken one after the other in a short period of time (therefore our method is suitable for hand-held device applications); 3) ghosting artifacts and misalignment are brought to a minimum. If a small misalignment between the two images occurs, our method is able to overcome it.

**Publication.** The work, presented in this chapter, has been published in the following paper: *H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, "High-dynamic-range image recovery from flash and non-flash image pairs." The Visual Computer: International Journal of Computer Graphics 33.6-8 (2017): 725-735. Won Best Paper Award at Computer Graphics International 2017, Yohohama, Japan.*

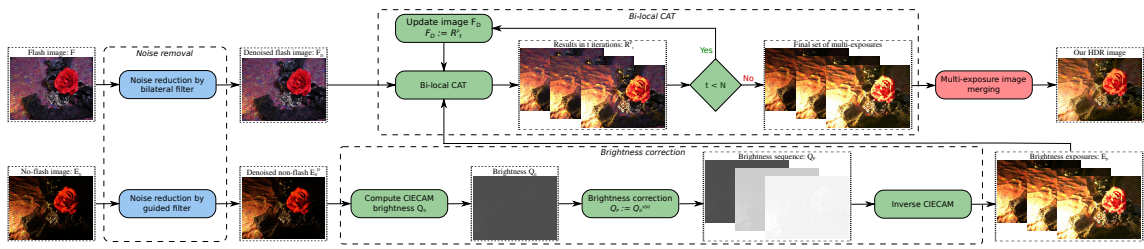
---

## 2 Related works

The entire dynamic range of a real-world scene cannot be captured by today's camera sensors. That is why, digital images of scenes with high-dynamic luminance range, are either under- or over-exposed. **The classical technique** for obtaining a high-dynamic range image without under-/over-exposed regions uses a set of images, taken at various exposure settings [114, 115]. Debevec et al. [114] first exploit the reciprocity property of imaging systems to construct the response curve of multi-exposure images and to recover their HDR radiance map. Reversely, Mann et al. [115] compute a floating-point image as a representation of an "undigital" image with an extended dynamic range, without any prior knowledge about the response curve of the imaging device. The floating point HDR image is yet again computed from a set of multi-exposure images. The general concept of using multi-exposure images for creating HDR images is highly exploited in today's photography. However, this approach has several main drawbacks, including possible image misalignment and ghosting for scenes with moving objects. To this end, there exists a number of techniques, designed to handle misalignment and ghosting artifacts [116], [117], [118], [119].

To overcome the main limitations of multi-exposure methods, Tocci et al. [120] propose an **optical architecture** which automatically captures three optically-aligned images at different exposures by splitting the light from a single lens and focusing it onto high, medium and low exposure imaging sensors. However, the proposed optical advancement is not available for massive use and its construction is costly. In contrast, other methods use a **single-coded image** to recover per-pixel exposures [121, 122]. They rely on a spatially varying optical mask on the sensor, giving different exposures to adjacent pixels. The coded exposures are mapped to an HDR image using reconstruction techniques, such as interpolation [122], piece-wise linear estimators, based on Gaussian mixture mod-





**Figure 10.1** – Main flowchart. In the noise removal step (blue boxes), we denoise the two input images. The brightness correction computes a sequence of multiple brightness images. Then, for each of the images in this sequence, we apply the bi-local CAT  $N$  times (by using the flash image) and we obtain the final multi-exposure images. They are merged into an HDR image in the last step (in red).

els [123], and the recently proposed sparse reconstruction, based on convolution sparse coding [124]. However, such reconstructions are computationally costly: they require hardware modification and they can introduce artifacts if the mask is regular and a simple interpolation is used.

Furthermore, a new method for **image brightening from a single image**, using standard digital cameras, has been recently introduced [125]. Li et al. create three virtually exposed images from a single image by increasing the brightness of the under-exposed regions. The brightness increase is carried out by a non-decreasing function in a newly designed “simplified” CIE Lab color space. Unlike our method, Li et al.’s method does not explicitly compute and modify the brightness of the original image. It is used to brighten dark objects in outdoor scenes as well as to create a tone-mapped version of an HDR image by fusing the three virtual exposures. A brightening approach from a single image would not give plausible results if the input image contains a significant number of under-/over-exposed pixels, for which no information can be recovered from a single image.

Furthermore, Mertens et al. [126] propose an **exposure fusion**, which merges a sequence of multi-exposure images into an image with extended luminance range, which can be directly displayed on an LDR screen (a tone-mapped image). The fusion is guided by series of metrics which ensure that only the well-exposed values of each exposure image are kept in the result. Unlike the exposure fusion, which combines several multi-exposure images into one enhanced image, two other methods introduce **image enhancement techniques for flash photography, relying on only two images**. The methods in [127, 128] exploit the properties of flash and non-flash image pairs for dark environments. They combine the non-flash ambient light with the details from the flash image using a bilateral-filter-based image decomposition. That way, they enhance the quality of the non-flash image. Unlike the HDR imagery, which provides a number of LDR outcomes (tone mappers), the methods in [127, 128] generate a single LDR image, which cannot be extended to an HDR image. Other methods also take two differently exposed images as an input. The method in [129] is applied between blurred and noisy image pairs for the purpose of image deblurring, whereas the methods in [130, 131] take differently exposed subsequent frames from a video sequence to reconstruct an HDR video.

Matsuoka et al. [132] also exploit the properties of the flash image, this time in the context of **HDR imagery**. To construct an HDR image, the authors integrate a sequence of multi-exposure images in the wavelet domain. Before merging the multi-exposure images,



two steps are performed. First, the flash image is used to find an alpha mask of shadow regions of the long exposure image. Second, a noise removal technique, guided by the flash image, is applied to denoise these shadow regions. Unlike our method, Matsuoka et al.'s method does not explicitly involve the flash image into the creation of the HDR image (no flash image information is transferred into the final HDR image). Furthermore, similarly to multi-exposure methods, Matsuoka et al.'s method requires a tripod to shoot the multi-exposure images and it is suitable only for static scenes.

---

### 3 Our method

In the present section, we introduce our method for computing an HDR image from two images - a flash image  $F$  and a non-flash image  $E_0$ . Figure 10.1 illustrates the main flowchart of our method. The proposed method starts with a noise removal step, yielding free of noise flash and non-flash images. The brightness of the noise-free non-flash image is then modified during a brightness correction step, at the end of which we obtain a sequence of **multiple brightness images**. The images in this sequence often contain under-/over-exposed pixels, *i.e.* lack scene details. In the next step, an iterative bi-local CAT, the missing details are recovered using information from the noise-free flash image. That way, we generate a **final sequence of multi-exposure images**, which we then merge together into an HDR image.

---

#### 3.1 Noise removal

Our method starts by denoising the flash and non-flash images. Even though the flash image is considered a reliable image, containing no/very little noise, the flash may introduce grainy noise. Therefore, we apply a bilateral filter with a small kernel size on the flash image to handle any possible noise. The bilateral filter behaves well for a well-lit images, such as flash images. In contrast, for images shot in dark environments without a flash, experiments show that the guided filter [133] performs better than both bilateral and cross-bilateral filters [8, 127]. Therefore, we apply the guided filter to denoise the non-flash image.

---

#### 3.2 Brightness gamma correction

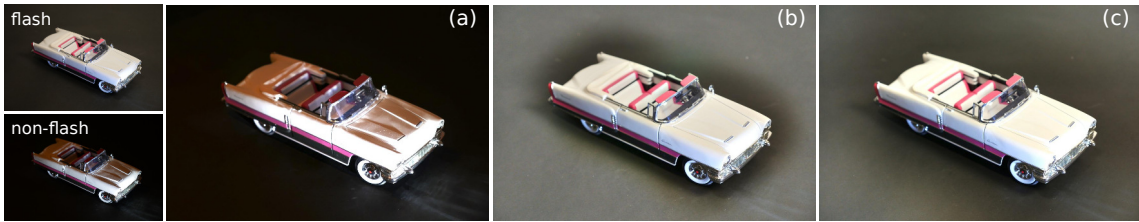
Creating an HDR image from multi-exposure images requires a knowledge of the CRF. Several methods for recovering the CRF exist [114, 115]. They recover the CRF up to a scale factor from at least two multi-exposure images. Once recovered, the CRF could be used to compute multi-exposure images from a single image.

To simplify the process of creating multi-exposure images, we no longer use prior knowledge about the CRF. Instead, we mimic the CRF by a brightness function. The brightness, which is one of the absolute color appearance attributes, is fundamental for our approach. It describes the intensity of the light source and its sensation depends on the adaptation to the scene light source. Furthermore, it varies with the environment (dark, dim, bright, etc.). A key advantage of the brightness over other color appearance attributes, such as lightness, is its unbounded range.

We compute the brightness  $Q_0$  of the non-flash image  $E_0$  using CIECAM02 [10]. The brightness  $Q_0$  is modified using a gamma correction function, where gamma is derived from a brightness dependent parameter  $p$ . By varying this parameter, we obtain multiple brightness images  $E_p$ . Their brightness  $Q_p$  is computed using the gamma correction, proposed by Bist et al. [134]:

$$Q_p = Q_0^{\gamma(p)}, \text{ where } \gamma(p) = 1 + \log\left(\frac{Q_{max}}{p}\right). \quad (10.1)$$

The gamma value  $\gamma(p)$  is obtained as a function of the correction parameter  $p$  and the maximum brightness  $Q_{max}$  of the non-flash image  $E_0$ . The parameter  $p$  is expressed in terms of  $Q_{max}$ . Therefore, we either increase (for  $p \leq Q_{max}$ ) or decrease (for  $p > Q_{max}$ ) the brightness of the non-flash image  $E_0$  to obtain each brightness exposure image  $E_p$ . The optimal choice of parameter  $p$  is discussed in section 3.5.



**Figure 10.2** – Images (a), (b) and (c) are obtained with a bi-local CAT, a local iCAM CAT and a local CAT, discussed in [7], respectively. The highly contrasting areas of the non-flash image can not be well-represented by a global illuminant and this is the main reason for the local CATs to fail when used in the context of flash/non-flash photography.

### 3.3 Iterative bi-local CAT

The brightness gamma correction, presented in the previous subsection, does not introduce new information and therefore, details in the under-/over-exposed areas of the non-flash image  $E_0$  cannot be recovered. To tackle the limitations of using a single image for the recovery of an HDR image, we consider an extra image - the flash image  $F$ . This image can easily be taken alongside the non-flash image in less than one second and contains reliable information about the shadows of the non-flash image as well as more scene details.

We propose a novel CAT, which carries out a transformation of the flash image  $F$  with respect to each image  $E_p$ . This transformation, that we call bi-local CAT, aims to adapt the colors of the image  $F$  to those of the image  $E_p$  as well as to remove the impact of the flash on the original scene lighting, while preserving the details of the image  $F$  (except for the flash shadows and reflections). Compared to previous works [127, 128], our method allows for an advanced combination of flash/non-flash light, color and detail, and at the same time is robust to small misalignment, flash shadows and reflections.

The bi-local CAT extends the local CAT, presented in the iCAM [21, 103]. The local iCAM CAT would compute a global illuminant for the image  $E_p$  and would locally adapt the colors of the flash image  $F$  to this illuminant. However, the wide luminance range of the image  $E_p$ , varying from pure black to pure white, cannot be correctly described

by a single illuminant. To transfer the high contrast areas of the image  $E_p$  onto the flash image  $F$ , the bi-local CAT computes a local representation of the illuminant of the image  $E_p$ , instead of a global one, as well as a local representation of the illuminant of the flash image  $F$ . Like standard CATs [10, 21], the bi-local CAT starts by converting the RGB stimuli of both images  $F$  and  $E_p$  into spectrally sharpened RGB signals [10]. Then, we apply the von Kries normalization pixel-wise to convert the spectrally sharpened RGB stimuli ( $R^F, G^F, B^F$ ) of the flash image into the adapted tristimulus responses ( $R_c, B_c, G_c$ ) as follows:

$$R_c = \left( \frac{R_w^{E_p}}{R_w^F} D + (1 - D) \right) R^F \quad (10.2)$$

$$G_c = \left( \frac{G_w^{E_p}}{G_w^F} D + (1 - D) \right) G^F \quad (10.3)$$

$$B_c = \left( \frac{B_w^{E_p}}{B_w^F} D + (1 - D) \right) B^F, \quad (10.4)$$

where the triples ( $R_w^{E_p}, G_w^{E_p}, B_w^{E_p}$ ) and ( $R_w^F, G_w^F, B_w^F$ ) are pixels from low-pass versions of the images  $E_p$  and  $F$  respectively (more details in the following paragraph). The adaptation factor  $D$  is given as follows [10, 21]:

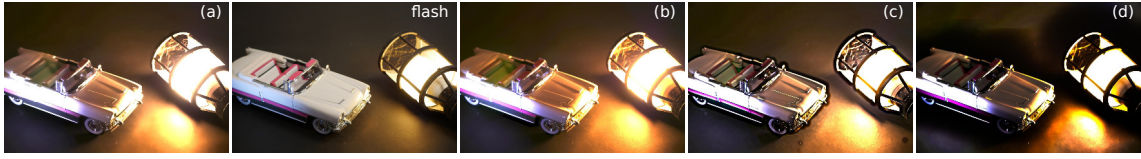
$$D = K \times S \times \left( 1 - \frac{1}{3.6} e^{\left( \frac{-L_A - 42}{92} \right)} \right), \quad (10.5)$$

where the scalar  $L_A$  is the adapting luminance and  $S$  is the surrounding factor, equal 1 in our method (as we carry out an adaptation of the colors of the flash image, and therefore, the surround is considered average). We use a coefficient  $K = 1$  to perform a full adaptation.

The von Kries normalization in equations (10.2), (10.3) and (10.4) computes the per-pixel ratio of two low-pass images ( $R_w^F, G_w^F, B_w^F$ ) and ( $R_w^{E_p}, G_w^{E_p}, B_w^{E_p}$ ), called white images (following the notation in [10]). So far, the von Kries normalization has been carried out either globally [20], in which case the white images boil down to white points, or locally between a single-point illuminant and a white image [7, 21] (more details are presented in part 2). To the best of our knowledge, a CAT has never been applied in a bi-local context. Figure 10.2 shows the advantage of the bi-local CAT over two local CATs for the purposes of this chapter.

The white images are computed directly from the flash image  $F$  and the image  $E_p$  as follows.

- The **flash white image** is computed by applying the guided filter. We observed that in our context the guided filter outperforms Gaussian and bilateral filters. Experiments show that Gaussian filter fails to transfer properly the shadows of the image  $E_p$ , introducing brand new shadow regions. Moreover, the bilateral filter introduces a lot of visible halo artifacts around the edges. In contrast, the guided filter suppresses the presence of such halo artifacts, preserves the shadow boundaries of the image  $E_p$  and robustly sharpens the details of the flash image, as shown in figure 10.3.
- The **white image of the image**  $E_p$  is the image  $E_p$  itself. The image  $E_p$  is obtained from the image  $E_0$ , to which we have applied the guided filter.



**Figure 10.3** – Image (a) is a brightness image  $E_p$ , for which  $p = 0.86Q_{max}$ . Images (b), (c) and (d) are results of applying our iterative bi-local CAT between image (a) and the flash image by using respectively a guided filter, a bilateral filter and Gaussian filter for obtaining the white image of image (a).

When applied iteratively, the bi-local CAT robustly adapts the colors of the image  $F$  to the colors of the image  $E_p$  and progressively removes flash shadows and reflections. The iterations are performed as follows:

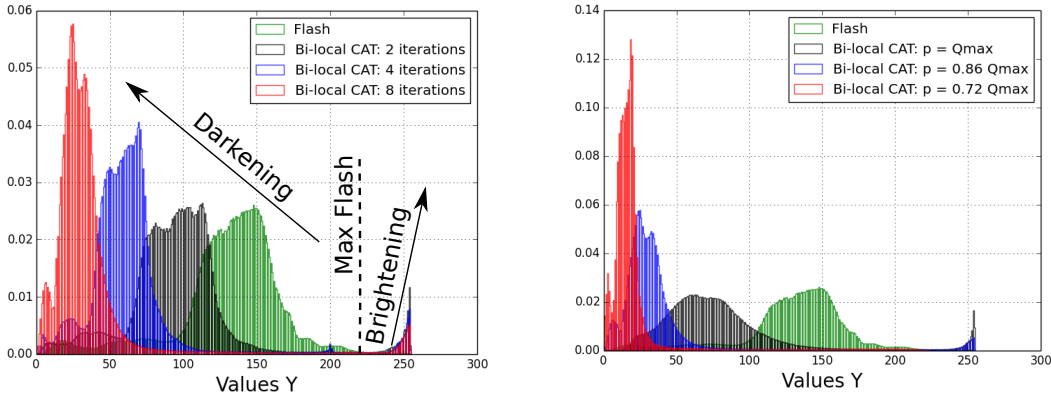
$$F_t^p = \begin{cases} biCAT(F, E_p), & \text{if } t = 1; \\ biCAT(F_{t-1}^p, E_p), & \text{if } t \in [2, N]. \end{cases} \quad (10.6)$$

During the first iteration ( $t = 1$ ), we carry out the bi-local CAT between the images  $F$  and  $E_p$ . For the following iterations, we perform the bi-local CAT between the result from the previous iteration  $F_{t-1}^p$  and the image  $E_p$ . After  $N$  iterations, we obtain the final exposure image  $F_N^p$ . During each iteration  $t$ , the flash white image is recomputed from the result  $F_{t-1}^p$ , whereas the white image of the image  $E_p$  remains unchanged. The two main properties of the iterative bi-local CAT are discussed hereafter.

**Property 1: Darkening.** When the ratio of the white images is less than 1, *i.e.*  $I_w^{E_p} / I_w^{F_t^p} < 1$ , where  $I$  stands for R, G and B channels, the bi-local CAT darkens the flash image  $F$  (left-hand plot in figure 10.4). As the white image of the image  $F$  is recomputed iteratively, the pixels of the flash image will keep decreasing until reaching an iteration  $k$ , for which the white image ratio becomes close to 1 (because the values  $(R_c, G_c, B_c)$  remain unchanged after the iteration  $k$ , see equations (10.2), (10.3) and (10.4)). To recover information in the under-exposed regions of the brightness multi-exposures, the maximum number of iterations does not have to exceed  $k$ . However, it still needs to be big enough for the bi-local CAT to transfer the scene ambience and remove flash shadows and reflections. More information on the optimal number of iterations is presented in section 3.5.

**Property 2: Brightening.** When the ratio of white images is greater than 1, *i.e.*  $I_w^{E_p} / I_w^{F_t^p} > 1$ , the bi-local CAT brightens the flash image (left-hand plot in figure 10.4). The pixel values of the flash image will keep increasing until reaching an iteration  $l$ , for which the white image ratio becomes close to 1. After the iteration  $l$ , the values of the flash image remain unchanged.

These two properties reveal the ability of the bi-local CAT to increase the dynamic range of the flash image  $F$  (by both darkening and brightening). They also reveal the importance of the brightness correction step in our algorithm. If we applied the iterative bi-local CAT only between the flash and the non-flash images (without computing multiple brightness images), we would progressively darken the values of the result  $F_t^p$  by shifting its histogram to the left (right-hand plot in figure 10.4). In this case, the final result would represent the brightness of the non-flash image  $E_0$  rather than the brightness of the scene. In contrast, once we obtain the sequence of multiple brightness images and



**Figure 10.4** – The left-hand plot shows luminance histograms of a flash image (in green) and results  $F_t^p$ . The bi-local CAT progressively darkens and brightens  $F_t^p$ , extending its range and contrast. The right-hand plot shows the influence of the parameter  $p$  on the transformation of  $F_t^p$  for  $t = 8$ . The smaller the value of  $p$ , the brighter the result  $F_t^p$  and the lesser the under-/over-exposed pixels. When  $p = Q_{max}$ , i.e. the result  $F_t^p$  is identical to the non-flash image, no brightening is performed (the flash pixels are progressively darkened).

perform the bi-local CAT, the histogram of the result  $F_t^p$  is shifted both to the left and to the right (we darken the pixels in the shadows and brighten the ones in the highlights).

### 3.4 Image fusion

The bi-local CAT yields a sequence of multi-exposure images, which are then merged together to recover an HDR image. We use Debevec et al.'s fusion method [114], which relies on a CRF estimation. In our method, we estimate the CRF from the final multi-exposure images.

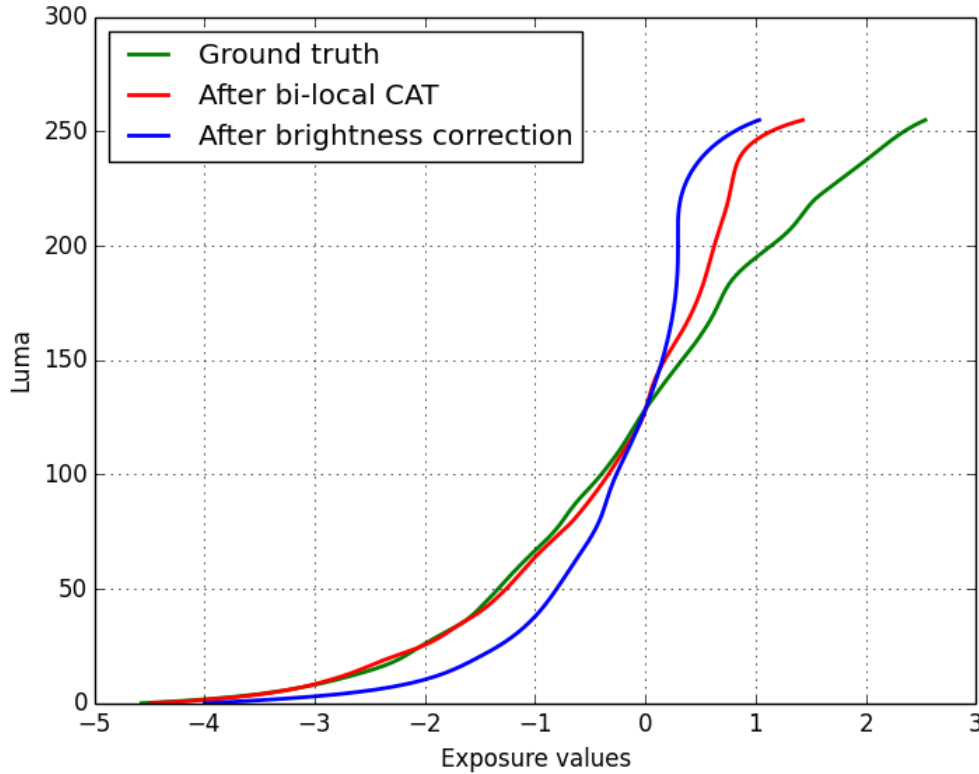
Additionally, we compute the real CRF from a sequence of real multi-exposure images to verify whether or not the CRF, used in our method, is similar to the real one. Figure 10.5 presents plots of the CRF, computed from final multi-exposure images, and the real CRF. We observe that the CRF, used in our method, approximates well the real CRF. This conclusion is based on several experiments, involving various real image sets. Figure 10.5 shows also the CRF, computed from the multiple brightness images. The CRF, estimated after the iterative bi-local CAT, is more accurate than the CRF, estimated after the brightness correction. This reveals a key advantage of our method over methods, based only on a brightness correction.

### 3.5 Choice of optimal values of $p$ and $N$

The efficiency of our method greatly depends on the parameter  $p$ , used in the brightness correction step. We analyze which values of  $p$  allow to compute a plausible approximation of the real CRF.

First, for every iteration  $t \in [1, 10]$  of the bi-local CAT, we compute multi-exposure images by using each value of  $p$  from the set  $\{(0.6 + 0.1i)Q_{max}\}_{i=0}^{39}$  (for a total of 400 final multi-exposure images). Experiments showed that values of  $p$  lower than  $0.6 \times Q_{max}$  result





**Figure 10.5** – The CRF, computed after the bi-local CAT, is highly accurate for exposure values less than 0 but tends to overestimate the luma for positive exposure values. This overestimation is due to the use of a flash image, which successfully captures details in shadows but may not manage to represent all the finest details, belonging to a light source (see figure 10.8).

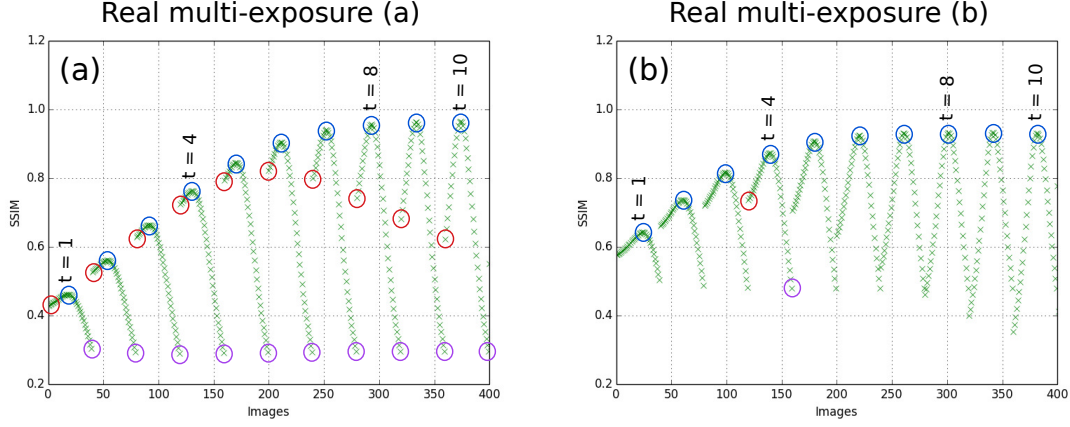
in over-exposure of the majority of pixels in the result and therefore, we exclude them. Second, we compute the structural similarity metrics (SSIM) [68] between each of the 400 multi-exposure images and each of several real multi-exposure images of the same scene (taken manually by a professional photographer). We observe a clearly defined peak, optimizing the SSIM value for each iteration  $t$  (figure 10.6). The peaks for all iterations  $t$  (per real multi-exposure image) correspond to the same  $p$ , which remains unchanged for all the different sets of real multi-exposure images, for which we performed this analysis. These sets of images were taken with two different types of cameras. Therefore, the value of  $p$  is also independent of the choice of camera. The value of  $p$  depends only on the exposure of the real multi-exposure image, but at the same time, it is insensitive to the choice of an exposure for the non-flash image.

We have experimentally derived the value  $p_i$  of the  $i^{th}$  final multi-exposure image as a function of  $Q_{max}$  and the image index  $i$ ,  $i \in \{1, \dots, M\}$ :

$$p_i = \left(1 - \frac{S \times C \times i}{10}\right) Q_{max}, \quad (10.7)$$

where  $C$  is a constant, which has experimentally been set to 0.7. The sign  $S$  is either equal to 1 for an increase of the non-flash brightness  $Q_0$ , or equal to -1 for a brightness





**Figure 10.6** – Plots (a) and (b) represent the SSIM scores (on the Y axis) between a real multi-exposure image and each of 400 multi-exposure images, obtained with our method. The 400 images are computed for each iteration  $t \in [1, 10]$  and each value of the parameter  $p$ , indexed by  $j \in [0, 39]$ . The X axis of each plot represents the number  $m \in [1, 400]$  of the final multi-exposure images, where  $m = t \times j$ . Each number  $m$  corresponds to an iteration  $t$  and a value of  $p$ . The curves in each plot correspond to the different iterations  $t$ . We find a clearly defined peak per iteration (circled in blue), optimizing the SSIM score. The highest value of  $p$  per curve (iteration) is circled in red, whereas the lowest value is circled in purple (analogically for the plot (b), where we give an example only for the forth iteration).

decrease. We have experimentally found out that the use of  $M = 6$  final multi-exposure images (out of which one is the non-flash image) helps generate HDR images, close to the ground truth. Our experiments have indicated that the exposure value  $X_i$  of the  $i^{th}$  final multi-exposure image can be expressed as  $X_i = X_0 + S \times i$ , where  $X_0$  is the exposure of the image  $E_0$ . The final multi-exposure images together with the computed exposure values allow to recover plausible HDR images.

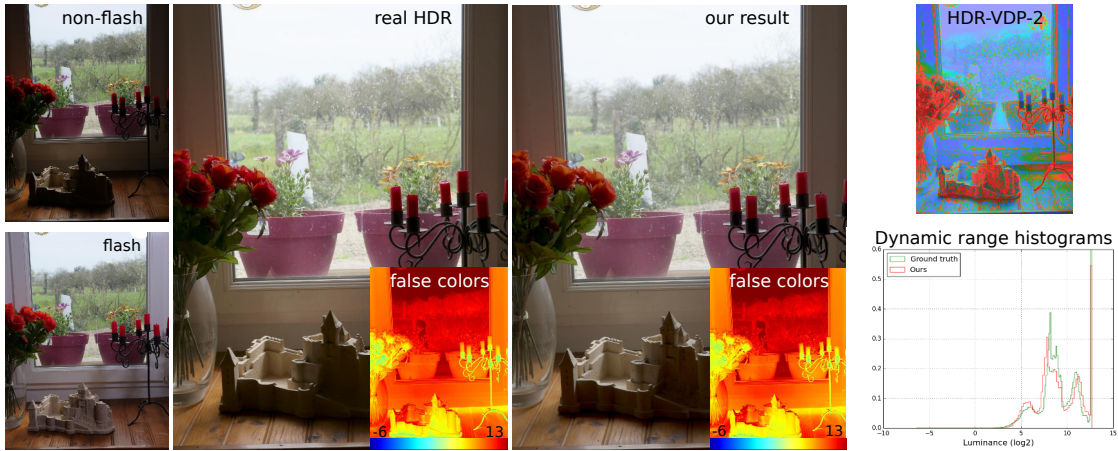
In our experiments, the maximum SSIM score was reached at the eighth iteration. We therefore chose to perform  $N = 8$  iterations of the bi-local CAT.

## 4 Results and evaluation

In this section, we present our HDR results and we evaluate their similarity to the ground truth.

**Experimental set up.** We have built a data set of images of real-world scenes, consisting of flash and non-flash images and real multi-exposure images. The flash and non-flash images were taken in a short period of time (less than one second) and were used to compute the results, shown in this chapter. Additionally, we took real multi-exposure images to recover a real HDR image per scene. A professional photographer has chosen the best exposure values in order to capture the finest details in the shadows and the highlights. To make the real HDR images representative of the ground truth, we used a tripod to avoid misalignment during the shooting process. We compare our results to the ground truth in the evaluation part of this section.

**Recommendations for the choice of non-flash images.** In our method, we choose the non-flash image  $E_0$  to be the lowest exposed image with less than 5% black pixels.

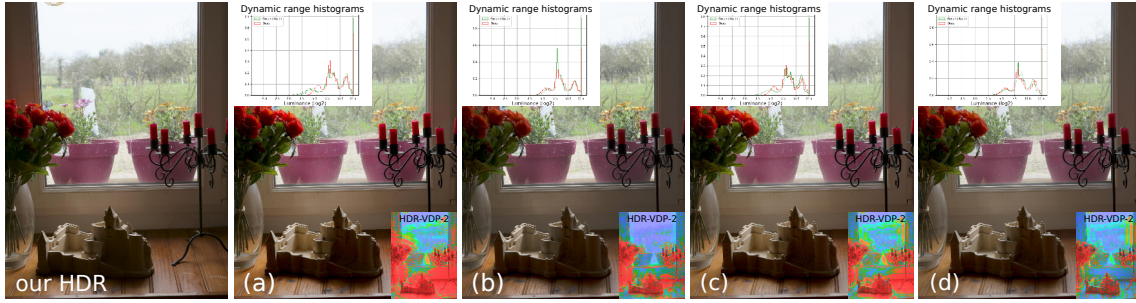


**Figure 10.7** – The real HDR image is computed by merging 5 real multi-exposure images. Our method produces an HDR image, the dynamic range of which is similar to that of the real HDR image (with a number of f-stops equal to 13). Moreover, the log2 luminance histograms show a similarity between the luminance distributions of our result and the real HDR image.

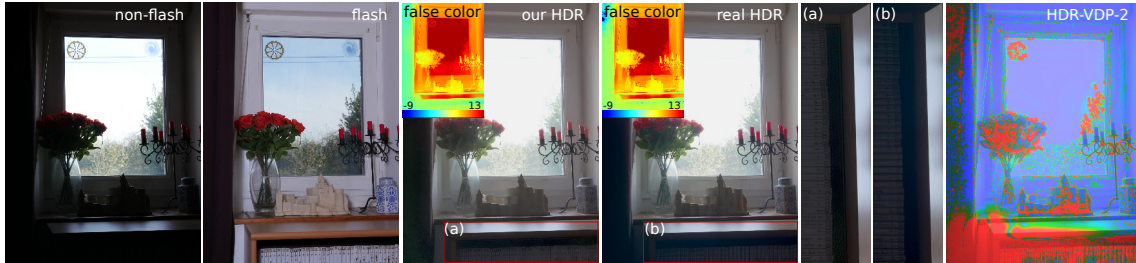
Despite the fact that the iterative bi-local CAT is able to recover the missing details in black pixel regions, in the case when the percentage of black pixels exceeds 5%, the non-flash image becomes too low-exposed and noisy. This results in a trade-off between the fidelity of the result and the successful noise removal when applying our method. The noisier the image, the bigger the kernel size of the guided filter and the greater the loss of details. The non-flash image  $E_0$  may not be the lowest exposed image for a given scene, however, its exposure time is still significantly short to allow for the flash and non-flash images to be taken subsequently without the use of a tripod.

**Evaluation.** Figure 10.7 presents an HDR result, obtained with our method, as well as a real HDR image of the scene. To evaluate the similarity between our HDR result and the real HDR image, we compute their luminance histograms (figure 10.7). Our method recovers the dynamic range of the real HDR images in our data set (resulting in the same number of f-stops as the real HDR images). The luminance distribution of our results is strongly correlated with the ground truth luminance. Moreover, we adopt the perceptual metrics HDR-VDP-2 [113] to visualize the perceptual difference between our HDR results and the ground truth. Red regions in the HDR-VDP-2 color-coded map indicate deviations from the ground-truth luminance. The color-coded maps in figure 10.7 reveal an overall high perceptual similarity between our result and the ground truth.

The real HDR images aim to represent the ground truth by merging a number of multi-exposure images. The more multi-exposure images we merge, the closer the HDR image is to the real-world scene. To show how close our results are to the ground truth, in figure 10.8 we compare them to several HDR images, obtained by combining 2, 3 and 5 real multi-exposure images. The HDR-VDP-2 metrics indicates that our HDR result is visually similar to the HDR image, computed from 5 real multi-exposure images. Moreover, the log2 luminance distribution of our HDR image is highly correlated with that of the real HDR image, obtained from 5 real multi-exposure images. In this sense, our result is closer to the ground truth than the HDR images, recovered by merging 2 and 3 real multi-exposure images.



**Figure 10.8** – Our HDR result is compared to four real HDR images, computed from 2 consequent, 2 non-consequent, 3 and 5 real multi-exposure images (images (a), (b), (c) and (d) respectively). The log2 luminance distribution of our result is similar to that of the HDR image, obtained by merging 5 real exposure images. The similarity is also reflected in the HDR-VDP-2 color maps. As the merge of 5 multi-exposure images represents better the ground truth than the merge of either 2 or 3 multi-exposure images, our method performs better than the classical multi-exposure methods, merging 2 and 3 multi-exposure images.



**Figure 10.9** – Result of applying our method to dark environment scenes with high dynamic range. Our method is able to recover most of the scene dynamics, as shown in the false color images. Snippets (a) and (b) visualize the most significant perceptual difference (indicated also by HDR-VDP-2) between the two HDR images. Our method recovers fine details from the flash image (the DVD labels), whereas the multi-exposure approach causes noise. Moreover, our method avoids the presence of ghosting artifacts like those in the real HDR image (the tree branches).

Despite the similarity with HDR images, computed from 5 real multi-exposure images, our results may differ from real HDR images at shadow areas. While adapting to the colors of the image  $E_p$ , our bi-local CAT preserves the details of the flash image in the shadows of the result. Reversely, taking low-exposures images in dark environments may cause noise in the shadows and compromise the integrity of the real HDR image. Figure 10.9 illustrates a key property of our method, *i.e.* the detail recovery. Our HDR image preserves the DVD labels in the shadows of the scene, unlike the HDR image, obtained from 5 real multi-exposures.

The main advantage of our method over the multi-exposure approach is illustrated in figure 10.10. The flash and non-flash images, shown in the figure, were taken with a hand-held camera, imitating a typical user case. Our method successfully recovers HDR images of non-still (slow moving) objects (such as people, posing for portraits) and avoids ghosting artifacts.

Finally, another advantage of our method consists of an automatic removal of soft shadows from the flash image, carried out by the bi-local CAT. If the flash image contains shadows, created by the flash, there is a risk that they will appear in the final result  $F_N^p$





**Figure 10.10** – Image (a) presents our result. All under-exposed pixels (below a threshold) of the non-flash image are shown in white in image (b). The under-exposed pixels are recovered in image (a) with the use of our iterative bi-local CAT. The flash and the non-flash images were taken with a hand-held camera, resulting in a small misalignment (the green circles). The misalignment is correctly handled by our bi-local CAT.



**Figure 10.11** – The flash shadows and reflections from the flash image have been automatically removed by our method.

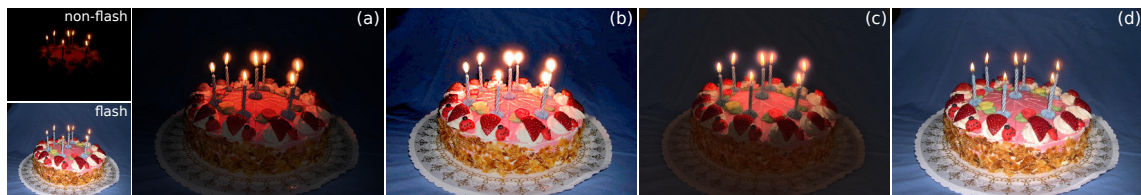
(and if they do, the result would look unnatural). It turns out, though, that 8 iterations of our bi-local CAT are enough to completely remove soft shadows from our HDR result, as illustrated in figure 10.11. Our method also reduces reflections, caused by the flash.

**Non-flash image enhancement.** Our method can be used in the context of non-flash image enhancement. We increase the quality of a non-flash image in terms of detail recovery and scene illumination enhancement, as shown in figure 10.12.

Given flash and non-flash images, we automatically recover an HDR image and then we use various tone-mapping operators to visualize it on an LDR screen. Figure 10.13 shows a comparison between our method and two state-of-the-art methods, all used in the context of non-flash image enhancement. Mertens et al. [126] fail to properly combine the flash and non-flash images, because the flash image is already well-exposed. Eisemann et al. [127] produce a single image as an outcome of their method. In contrast, our method provides a number of enhanced images, each resulting from a different tone-mapping operator.

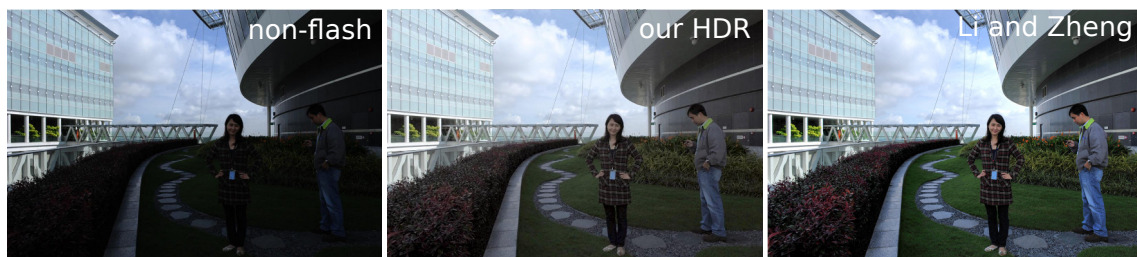


**Figure 10.12** – Non-flash image quality enhancement. The gamma correction of the non-flash image reveals the missing details on the top-right and lower-right corners. These details are recovered with our method using the flash image (best viewed on screen).



**Figure 10.13** – Images (a) and (b) are our results, obtained respectively by fusing the final multi-exposure images (with the method in [126]), and by using the tone-mapper in [9] on the reconstructed HDR image. Image (c) is Eisemann et al.'s result [127], whereas image (d) is Mertens et al.'s result [126]. The flash and non-flash images are courtesy of [127].

As a remark, Eisemann et al. [127] combine flash details and non-flash lighting using two-parameter-dependent bilateral filter in order to enhance the quality of the non-flash image. Before merging the two input images, the authors increase the contrast of the non-flash image. However, this increase appears random. In contrast, we model the increase of the brightness of the non-flash image by a gamma curve. The use of such a model is beneficial for obtaining naturally looking results.



**Figure 10.14** – Our tone-mapped HDR result is obtained directly from the non-flash image (courtesy of [125]). In the case of outdoor scenes, our method boils down to a single image brightening. Li et al. [125] have previously proposed a single-image-based brightening algorithm. The result of their fusion is presented in the third image.

**Limitations.** Our method uses flash/non-flash image pairs to acquire the dynamic range of real-world scenes. Due to the limited reach of the flash of today's cameras, our method is not suitable for outdoor scenes. However, in the case of outdoor scenes, containing reliable information in the shadows, we can apply the first step of our algorithm, the brightness correction, and that way, we are able to obtain an HDR image, as shown in figure 10.14. Although our method is not specially designed for handling HDR image



creation from a single image, our brightening step performs fairly, compared to the single image brightening, proposed in [125]. Our method is able to represent the dynamic range of the outdoor scene by modifying the brightness of the non-flash image, as illustrated by the tone-mapped HDR image in figure 10.14. On the other hand, Li et al.'s fusion method captures better the vividness of the scene.

---

## 5 Conclusion

In this chapter, we have proposed a novel method for creating HDR images, relying on only two images as an input - flash and non-flash images. We have automatically computed multiple exposure images by brightening the non-flash image and bi-locally adapting the colors of the flash image to the brightened non-flash image. By merging the so-computed multi-exposure images we have created HDR images. Our HDR results are similar to the HDR images obtained by merging five manually taken multi-exposure images. Our method is very efficient for the challenging dark environment scenes, which often contain noise and lack reliable information. Moreover, our method can be applied in the context of non-flash image enhancement and in comparison with existing methods, it provides various enhancement options. Due to the limited reach of the flash, our method is not robust for outdoor scenes, which are left for future work.

Furthermore, in this chapter, we introduced our novel CAT method, *i.e.* the bi-local CAT, and demonstrated its potential. When applied in the context of flash/non-flash image editing, the bi-local CAT successfully recovers details from the flash image, transfers non-flash image shadows and removes flash shadows and reflections. In the following part, we exploit the potential of the bi-local CAT for various example-guided applications, *e.g.* image de-noising, image de-blurring, texture transfer, etc.





## **Part IV**

### **Perceptual-guided filtering for image enhancement and image editing**



# Introduction

In the previous part, we introduced two new CAT algorithms, *i.e.* the cluster-based local CAT and the bi-local CAT. The cluster-based local CAT is an extension to our local CAT [7] for HDR images. It help adapt the colors of an HDR image to the illuminant of another HDR image. The cluster-based local CAT is applied directly to the HDR domain. In contrast, the bi-local CAT is applied between two LDR images (of the same scene) and it helps recover an HDR image from these LDR images (chapter 10). When we perform the bi-local CAT between flash and non-flash images (of the same scene), we recover fine details from the flash image without compromising the ambience of the original scene (*i.e.* the non-flash lighting).

In this part, we exploit the potential of the bi-local CAT for various image editing applications such as image de-noising, image de-blurring, texture transfer, etc. We propose a novel guidance-based filter in which we embed the bi-local CAT, as presented in chapter 11. The proposed filter performs as good as (and for certain applications even better than) state-of-the art methods.



# Multi-purpose CAT-based guidance filter 11

---

## 1 Introduction

Many image processing applications require image smoothing techniques for noise reduction. Classical filters, such as Gaussian filter and median filter, often blur edges in the smoothing process. Edge-preserving filters like the bilateral filter [8] also experience a trade-off between the noise removal and the image integrity, as they use a single image to build their kernel.

In contrast, guidance-based filters incorporate additional information into the filtering process through the use of a guidance image. The guidance image, which is often a noise-free sharp image, is used explicitly in the estimation of the filter kernel. That way, the smoothing of the input image is carried out more efficiently and the amount of information loss is reduced. The guidance-based filters are commonly applied for example-based noise reduction [127, 128, 133], depth-map filtering [135], image matting [133], etc.

Local optimizations in the guidance-based filters may concentrate blur around sharp edges and cause a decrease in the sharpness of the output image. The guidance-based filters may compromise the input lighting atmosphere by smoothing down input reflections. Furthermore, they do not recover details from the guidance image, which limits their applicability to texture transfer and detail enhancement.

To tackle some of the limitations of the guidance-based filters, we present a new guidance filter, based on color appearance and color perception [10]. Our filter carries out a patch-wise linear transformation between an input image and a guidance image. In practice, this linear transformation adds details from the guidance image to a low-pass version of the input image. The amount of added details is controlled by a scaling coefficient in which we embed our new bi-local CAT, presented in chapter 10. The bi-local CAT strongly contributes to preserving input scene details such as input reflections. Along with preserving the input scene ambience, the embedded bi-local CAT plays a major role in transferring details from the guidance image.

In comparison to existing guidance-based filters, our CAT-based filter provides solutions to various digital imaging problems. In this chapter, we address several image processing applications, *i.e.* image denoising, texture transfer, detail enhancement with near-to-infrared (NIR) images, image deblurring, mask refinement and skin beautification. Our results compare fairly to results from state-of-the-art methods. Furthermore, our filter outperforms existing guidance-based filters in terms of image sharpness, detail enhancement and preservation of the input lighting.

The rest of the chapter is organized as follows. Existing guidance-based filters along with their advantages and limitations are discussed in section 2. Section 3 introduces our



**Table 11.1** – Notations.  $\Omega$  denotes the set of spatial coordinates of the input, guidance and output images.

Notation	Definition
$\mathbf{p} : \Omega \subset \mathbf{R}^2 \rightarrow \mathbf{R}^m$	input image
$\mathbf{g} : \Omega \subset \mathbf{R}^2 \rightarrow \mathbf{R}^n$	guidance image
$\mathbf{q} : \Omega \subset \mathbf{R}^2 \rightarrow \mathbf{R}^m$	output image
$\mathbf{J} \in \{\mathbf{p}, \mathbf{g}, \mathbf{q}\}$	$\mathbf{p}$ , $\mathbf{g}$ or $\mathbf{q}$
$N$	resolution of $\mathbf{J}$
$\mathbf{J}_i$	$i^{th}$ pixel of $\mathbf{J}$ ( $i = \{1, \dots, N\}$ )
$\mathbf{J}^c$	$c^{th}$ channel of $\mathbf{J}$ ( $c = \{1, 2, 3\}$ )
$\mathbf{J}^W$	white image (low-pass version) of $\mathbf{J}$
$\mathbf{J}^W(c)$	$c^{th}$ channel of $\mathbf{J}^W$
$\omega_k$	$k^{th}$ image patch (window) of a given size
$\mu_{\mathbf{J}_k}$	mean vector of $\mathbf{J}$ in the patch $\omega_k$

CAT-based guidance filter. Applications and results are presented in section 5. Finally, the last section concludes the chapter.

## 2 Related work

In the following section, we present guidance-based image filters and we discuss their objectives and functionalities as well as their limitations. The input of guidance-based filters consists of an input image  $\mathbf{p}$  and a guidance image  $\mathbf{g}$ . The main goal of guidance-based filters is to filter the input image  $\mathbf{p}$  using information from the guidance  $\mathbf{g}$ . Depending on how the information of  $\mathbf{g}$  is incorporated into the filtering process, the guidance-based filters can be classified into two categories: convolution-based guidance filters and model-based guidance filters. Notations, used in rest of the chapter, are given in table 11.1.

### 2.1 Convolution-based guidance filters

Convolution-based guidance filters compute an output image  $\mathbf{q}$  in a patch-wise manner. Each pixel  $\mathbf{q}_k$  of the output image  $\mathbf{q}$  is computed as a weighted mean of its neighboring pixels in the input image  $\mathbf{p}$ , *i.e.* the pixels, belonging to a window  $\omega_k$ , centered at  $\mathbf{p}_k$ :

$$\mathbf{q}_k = \sum_{j \in \omega_k} w_{kj}(\mathbf{g}) \mathbf{p}_j. \quad (11.1)$$

The weights  $w_{kj}$  are defined as functions of the guidance  $\mathbf{g}$ . Hereafter we present two convolution-based guidance filters: the cross-bilateral filter and the rolling guidance filter.

### 2.1.1 Cross-bilateral filter

**Definition.** The cross-bilateral filter [127] (introduced as joint bilateral filter in [128]) defines the weights  $\omega_{kj}$  from equation 11.1 as follows:

$$w_{kj}(\mathbf{g}) = \frac{1}{C_k} \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_j\|^2}{2\sigma_r^2}\right) \exp\left(-\frac{\|\mathbf{g}_k - \mathbf{g}_j\|^2}{2\sigma_c^2}\right). \quad (11.2)$$

The vector  $\mathbf{x}_i$  consists of the spatial coordinates of the  $i^{th}$  pixel. The standard deviations  $\sigma_r$  and  $\sigma_c$  control respectively the spatial and the color similarities between the pixels. The coefficient  $C_k$  is a normalizing coefficient, ensuring that  $\sum_{j \in \omega_k} w_{kj} = 1$ .

The weights of the cross-bilateral filter depend on the guidance image  $\mathbf{g}$  and not on the input image  $\mathbf{p}$  like in the bilateral filter [8], as shown in equation 11.2. If the input image  $\mathbf{p}$  and the guidance  $\mathbf{g}$  are identical, the cross-bilateral filter boils down to the bilateral filter. When used for denoising, the bilateral filter presents a trade-off between the integrity of the results and the noise removal, as the weights  $w_{kj}$  depend on the input image. In contrast, the cross-bilateral filter robustly denoises a noisy input image using a sharp noise-free image as a guidance. The cross-bilateral filter can also be used in the context of depth refinement for filtering a corrupted depth map with the use of an RGB image [135].

**Limitations.** Hereafter, we discuss several limitations of the cross-bilateral filter:

1. The cross-bilateral filter may cause the appearance of gradient reversal artifacts around the edges of the results, as illustrated in [133].
2. The cross-bilateral filter cannot recover details from the guidance, as the output is a weighted mean of the input image (equation 11.1). To this end, the filter is mainly applied in the context of image denoising.

### 2.1.2 Rolling guidance filter

The rolling guidance filter [136] is an iterative filtering approach, which uses a dynamic guidance image to filter the input image  $\mathbf{p}$ . The initial guidance image  $\mathbf{g}^{(0)}$  is computed by applying Gaussian filter to the input image  $\mathbf{p}$ . During each iteration  $k$ , the outcome  $\mathbf{q}^{(k)}$  is obtained with the cross-bilateral filter given the input image  $\mathbf{p}$  and the outcome  $\mathbf{q}^{(k-1)}$ , resulting from the previous iteration (*i.e.*  $\mathbf{q}^{(k-1)}$  becomes the guidance image  $\mathbf{g}^{(k)}$ ). The rolling guidance filter is suitable for a number of applications, *e.g.* texture removal and separation, image abstraction, gradient detection, etc. However, as the guidance image in the rolling filter depends on the input image  $\mathbf{p}$ , the rolling filter has a limited impact on applications such as texture transfer, image denoising, non-flash image enhancement, etc.

## 2.2 Model-based guidance filters

Model-based guidance filters assume the existence of a parametric model between the output  $\mathbf{q}$  and the guidance  $\mathbf{g}$ . In the following sections, we discuss the models, used in the guided filter, the weighted guided filter and several distribution-based transformations between  $\mathbf{p}$  and  $\mathbf{q}$ .

### 2.2.1 Guided filter

**Definition.** The guided filter assumes that there exists a linear model between patches of the output  $\mathbf{q}$  and the guidance  $\mathbf{g}$ . This assumption is supported by observations in [133], showing that the color distribution of image patches forms a line in the RGB space. The color line model [137] has previously been adopted for the purposes of image matting [138], image denoising [139] and super resolution [140].

Rather than computing a weighted mean of the input pixels in a patch, like in the cross-bilateral filter, the guided filter builds a linear model between corresponding patches of the guidance  $\mathbf{g}$  and the output  $\mathbf{q}$  as follows:

$$\mathbf{q}_i^c = \mathbf{a}_k^T \mathbf{g}_i + b_k, \quad (11.3)$$

where the scalar  $\mathbf{q}_i^c$  stands for one of the three channel values of the output pixel  $\mathbf{q}_i$ , the coefficient  $\mathbf{a}_k$  is a vector and the coefficient  $b_k$  is a scalar. The linear coefficients  $\mathbf{a}_k$  and  $b_k$  are determined by minimizing the sum of quadratic errors between the channel values  $\mathbf{p}_j^c$  and  $\mathbf{q}_j^c$ , belonging to each window  $\omega_k$ , centered at pixel  $\mathbf{p}_k$ :

$$\begin{aligned} (\hat{\mathbf{a}}_k, \hat{b}_k) &= \underset{\mathbf{a}_k, b_k}{\operatorname{argmin}} \sum_{j \in \omega_k} (e_j^c)^2 \\ &= \underset{\mathbf{a}_k, b_k}{\operatorname{argmin}} \sum_{j \in \omega_k} \left( (\mathbf{p}_j^c - (\mathbf{a}_k^T \mathbf{g}_j + b_k))^2 + \epsilon \|\mathbf{a}_k\|^2 \right), \end{aligned} \quad (11.4)$$

where the regularization term  $\epsilon \|\mathbf{a}_k\|^2$  prevents  $\mathbf{a}_k$  from becoming too big. By assuming that the sum of quadratic errors  $\sum_{j \in \omega_k} (e_j^c)^2$  is normally distributed, the solution for the linear coefficients  $\mathbf{a}_k$  and  $b_k$  is given by linear regression. Both coefficients, the vector  $\hat{\mathbf{a}}_k$  and the scalar  $\hat{b}_k$ , are determined for each input channel  $\mathbf{p}^c$  and for each window  $\omega_k$ , centered at  $\mathbf{p}_k$ :

$$\hat{\mathbf{a}}_k = (\Sigma_k + \epsilon \mathbf{U})^{-1} \operatorname{cov}(\mathbf{p}_k^c, \mathbf{g}_k) \quad (11.5)$$

$$\hat{b}_k = \mu_{\mathbf{p}_k^c} - \hat{\mathbf{a}}_k^T \mu_{\mathbf{g}_k}, \quad (11.6)$$

where  $\operatorname{cov}(\mathbf{p}_k^c, \mathbf{g}_k)$  is a vector, consisting of the covariances in the window  $\omega_k$  between the input channel  $\mathbf{p}^c$  and each channel of  $\mathbf{g}$ . The matrix  $\Sigma_k$  is the covariance matrix of  $\mathbf{g}$  in  $\omega_k$ , whereas  $\mathbf{U}$  is the identity matrix. The scalar value  $\mu_{\mathbf{p}_k^c}$  is the mean of  $\mathbf{p}^c$  in  $\omega_k$ , whereas the vector  $\mu_{\mathbf{g}_k}$  is the mean of  $\mathbf{g}$  in  $\omega_k$ . The parameter  $\epsilon$  determines which edges shall be smoothed and which shall be preserved. Patches with variance less than  $\epsilon$  are smoothed, whereas those with variance higher than  $\epsilon$  are preserved.

The final coefficients  $\bar{\mathbf{a}}_i$  and  $\bar{b}_i$ , used to compute the output channel value  $\mathbf{q}_i^c$ , are obtained by averaging the values of  $\hat{\mathbf{a}}_k$  and  $\hat{b}_k$  for all the windows  $\omega_k$ , covering the input pixel  $\mathbf{p}_i$ . Once the final coefficients  $\bar{\mathbf{a}}_i$  and  $\bar{b}_i$  are determined, the pixels of the output image  $\mathbf{q}$  are computed using the affine transform, given in equation 11.3.

The guided filter is mainly applied for image denoising, image sharpening and image feathering [133]. The guided filter avoids gradient reversal artifacts (unlike the cross-bilateral filter) and it keeps the image edges intact. From equation 11.3 we derive:  $\nabla \mathbf{q}_i^c = \mathbf{a}_k^T \nabla \mathbf{g}_i$ , which shows that the guided filter preserves the gradient of the guidance image.

**Limitations.** Despite the benefits of the guided filter in many image processing applications, the guided filter has several main drawbacks, as discussed hereafter:



**Figure 11.1** – Four results of applying guided filter in the context of image denoising. The increase of the window size  $r$  influences the integrity of the output: the shadow boundaries change, the illumination is flattened (which decreases the contrast), the input light reflections are smoothed, *i.e.* the input illumination is compromised. With the increase of  $r$  more details are recovered in the output and more noise is removed. However, the ambience of the scene is compromised.

1. In flat areas of the guidance image:  $\mathbf{g}_i = \mu_{\mathbf{g}_k}$  and  $\Sigma_k = [0]$ . Then, when the window size  $r$  is small,  $cov(\mathbf{p}_k^c, \mathbf{g}_k) \approx \vec{0}$  for each input channel  $\mathbf{p}^c$ . This means that  $\hat{\mathbf{a}}_k \approx \vec{0}$  and the output pixel  $\mathbf{q}_i$  is computed by averaging the input pixels in  $\omega_k$ , *i.e.*  $\mathbf{q}_i \approx \mu_{\mathbf{p}_k}$ . Therefore, when  $r$  is small, input patches, corresponding to flat patches in the guidance image, are smoothed by the guided filter. This results in a low depth of field of the output images (see figure 11.1,  $r = 4$  and  $r = 8$ ). In contrast, when  $r$  increases,  $\mathbf{a}_k$  also increases. In this case, the guided filter preserves more details from the guidance image. Although this may work well for image denoising, the increase of the window size  $r$  compromises the integrity of the input lighting. As shown in figure 11.1, the input non-flash reflections are not present (as they are smoothed) in the output images, obtained with  $r = 16$  and  $r = 32$ . Moreover, undesired flash reflections appear on the pots in the result, obtained with  $r = 32$ .
2. The assumption for a local affine transform between  $\mathbf{g}$  and  $\mathbf{q}$  does not hold if two or more dominant colors are present in a patch. If this is the case, the performance of the guided filter is compromised.
3. The computation of the mean  $\mu_{\mathbf{p}_k^c}$  and the mean vector  $\mu_{\mathbf{g}_k}$  in the window  $\omega_k$  is performed through a simple averaging in the window  $\omega_k$  of the values of the input channel  $\mathbf{p}^c$  and the pixels of the guidance  $\mathbf{g}$  respectively.
4. The final regression parameters  $\mathbf{a}_i$  and  $b_i$ , used to compute the output channel value  $\mathbf{q}_i^c$ , are obtained by averaging the parameters  $\mathbf{a}_k$  and  $b_k$  for all the windows  $\omega_k$ , for which  $\mathbf{p}_i \in \omega_k$ .
5. The guided filter does not recover details from the guidance image in the input shadow patches. In the input shadow areas, we have  $\hat{\mathbf{a}}_k = \vec{0}$  and  $\mathbf{q}_i = \mu_{\mathbf{p}_i}$  (see equations 11.5 and 11.3), *i.e.* the value of a pixel in an input shadow area is an average of its neighboring pixels in the input image.

To tackle some of the limitations of the guided filter, Li et al. [141] have extended the guided filter to a weighted guided filter, which is discussed in the following section.

### 2.2.2 Weighted guided filter

Similarly to the guided filter, the key idea of the weighted guided filter lies in the existence of a local linear model between the guidance image  $\mathbf{g}$  and the output  $\mathbf{q}$ . The weighted guided filter incorporates an edge-aware weighting function  $\Gamma(\mathbf{g})$  in the minimization from equation 11.4 as follows:

$$(\hat{\mathbf{a}}_k, \hat{b}_k) = \underset{\mathbf{a}_k, b_k}{\operatorname{argmin}} \sum_{j \in \omega_k} \left( (\mathbf{p}_j^c - (\mathbf{a}_k^T \mathbf{g}_j + b_k))^2 + \frac{\epsilon}{\Gamma(\mathbf{g})} \|\mathbf{a}_k\|^2 \right). \quad (11.7)$$

The weighting function  $\Gamma(\mathbf{g})$  is a regularization function, which ensures that, at an edge, the components of the vector  $\hat{\mathbf{a}}_k$ , computed with the weighted guided filter, are closer to 1 than those, computed with the guided filter. By preserving the sharp edges of the guidance image, the weighted guided filter aims to overcome one of the main limitations of the guided filter, *i.e.* the decrease of the depth of field. However, the weighted guided filter exhibits similar drawbacks to those of the guided filter. This is due to the common assumption in both filters, *i.e.* the existence of a linear model between  $\mathbf{q}$  and  $\mathbf{g}$  and to the common framework, *i.e.* the minimization, which boils down to a linear regression.

Furthermore, there exist linear models assuming that the input image  $\mathbf{p}$  and the guidance  $\mathbf{g}$  follow a particular distribution. In part II, we have already presented the state-of-the-art distribution-based transformations. The following section introduces them from a different perspective, *i.e.* in the context of patch-wise image distribution filters.

### 2.2.3 Distribution-based transformations

The distribution-based filters carry out a linear transformation of statistical features of two distributions. Before delving into the details of the distribution-based filters, we first analyse the model from equation 11.3. Equation 11.3 is a patch-wise linear model between the channels  $\mathbf{p}^c$  and  $\mathbf{q}^c$ . To incorporate all three channels of  $\mathbf{p}$  and  $\mathbf{q}$  into a single linear model, we express  $\mathbf{q}_i$  as follows:

$$\mathbf{q}_i = \mathbf{a}_k^T \mathbf{g}_i + \mathbf{b}_k, \quad (11.8)$$

where  $\mathbf{a}_k$  is now a matrix and  $\mathbf{b}_k$  is a vector,  $\mathbf{q}_i$  and  $\mathbf{g}_i$  are vectors.

Solutions for  $\mathbf{a}_k$  and  $\mathbf{b}_k$  can be given with a linear regression by minimizing the quadratic norm of the difference  $\mathbf{p} - \mathbf{q}$ :

$$\hat{\mathbf{a}}_k = \Sigma_k^{-1} \operatorname{cov}(\mathbf{p}_k, \mathbf{g}_k) \quad (11.9)$$

$$\hat{\mathbf{b}}_k = \mu_{\mathbf{p}_k} - \hat{\mathbf{a}}_k^T \mu_{\mathbf{g}_k}. \quad (11.10)$$

The coefficients  $\hat{\mathbf{a}}_k$  and  $\hat{\mathbf{b}}_k$  are determined for each vector  $\mathbf{p}_k$  as opposed to each channel value  $\mathbf{p}_k^c$  (like in the guided filter).

We now replace the coefficient  $\mathbf{b}_k$  in equation 11.8 with its solution from equation 11.10 and we obtain:

$$\mathbf{q}_i = \mathbf{a}_k^T (\mathbf{g}_i - \mu_{\mathbf{g}_k}) + \mu_{\mathbf{p}_k}. \quad (11.11)$$

The linear model 11.11 presents a general linear transformation of statistical features between two images  $\mathbf{p}$  and  $\mathbf{g}$ . Solutions for the coefficient  $\mathbf{a}_k$  (other than the solution in

equation 11.9) can be derived by assuming an MGD for the distributions of  $\mathbf{p}$  and  $\mathbf{g}$ . The Gaussian assumption is often valid for the distributions of entire images or clusters of images. Therefore, the distribution-based transformations are performed between entire images or clusters of images, rather than in a patch [3, 7, 34]. In part II, we have illustrated that a certain class of images (colors-based style images) can be divided into color clusters following the MGD.

The Gaussian assumption helps represent  $\mathbf{a}_k$  as a linear mapping between the covariance matrices of  $\mathbf{p}$  and  $\mathbf{g}$ . In the simple case of independent color space channels,  $\mathbf{a}_k$  is a diagonal matrix with diagonal elements, computed as the ratio of the standard deviation of  $\mathbf{g}^c$  to the standard deviation of  $\mathbf{p}^c$  [3]. In the case of dependent color space channels, the solution for  $\mathbf{a}_k$  can admit a number of solutions, *e.g.* Cholesky decomposition, square root decomposition and Monge-Kantorovich closed-form solution (for discussion and comparison, see part II and [34]). Cholesky decomposition and square root decomposition give exact but not optimal solutions for  $\mathbf{a}_k$ . In contrast, the Monge-Kantorovich closed-form mapping is optimal, as it is derived as a solution to the Monge-Kantorovich optimization problem [63].

The distribution-based filters transform one MGD into another one. To this end, they are usually applied in the context of color transfer between two images of different scenes [3, 34]. As discussed in part II, the plausibility of the results, obtained with a color transfer method, depends on the Gaussian assumption. In cases when the Gaussian assumption is too strong, local color transfer methods [6, 7] carry out the linear transformation 11.11 between image clusters. For comparison purposes, in section 5 we perform the Monge-Kantorovich transformation between image patches rather than image clusters and we apply it in the context of image denoising.

## 2.3 Discussion

Depending on the initial assumptions and goals, the existing guidance-based filters propose various solutions for the coefficient  $\mathbf{a}_k$  from equation 11.11. However, these solutions have certain limitations. For example, the linear model in the guided filter may fail when two or more dominant colors are present in a patch. Moreover, both guided and cross-bilateral filters may compromise the integrity and the depth of field of the input image through smoothing input edges and reflections. Finally, the performance of the distribution-based transformations strongly depends on the Gaussian assumption, which, in some cases, may be too restrictive. In the next section we introduce our novel guidance filter, which provides a new solution for  $\mathbf{a}_k$ , based on color perception through a chromatic adaptation model. Our guidance filter aims to overcome the main limitations of the existing guidance-based filters.

# 3 Our method

As presented in the previous section, most of the guidance-based filters are built under a particular assumption. For example, the guided filter relies on a local optimization, whereas the Monge-Kantorovich mapping is distribution-dependent. These assumptions may compromise the performance of the given filter.



To address the main limitations (discussed in section 2) of the existing guidance-based filters, we propose a new guidance filter, based on color perception. Our filter does not perform a local optimization and it is distribution independent. The core of our filter is our new bi-local CAT. We embed the bi-local CAT in a linear transformation between an input image  $\mathbf{p}$  and a guidance image  $\mathbf{g}$ . In practice, our CAT-based filter transfers a well-defined amount of details from the guidance image to a low-pass version of the input image. The amount of transferred details is controlled by the bi-local CAT. This makes our filter suitable for a number of applications, such as texture transfer and detail enhancement (more applications are presented in section 5). The core of our method, *i.e.* the bi-local CAT has been introduced in chapter 9 in part III. Before delving into the details of the CAT-based filter, we present the connection between the bi-local CAT and the guidance-based filters.

### 3.1 Bi-local CAT in the frame of guidance-based filters

In part III, we extended the local CAT by performing a von Kries normalization between two images, rather than between an image and an illuminant. The bi-local CAT between images  $\mathbf{p}$  and  $\mathbf{q}$ , which admits the following form:

$$\mathbf{q}_i^c = \left( \frac{\mathbf{p}_k^W(c)}{\mathbf{g}_k^W(c)} D + (1 - D) \right) \mathbf{g}_i^c, \quad (11.12)$$

where  $\mathbf{p}^W$  and  $\mathbf{g}^W$  are low-pass representations of the input image  $\mathbf{p}$  and the guidance image  $\mathbf{g}$  respectively, *i.e.*  $\mathbf{p}_k^W$  and  $\mathbf{g}_k^W$  are the mean pixel values of  $\mathbf{p}$  and  $\mathbf{q}$  in a window  $\omega_k$ , centered at  $\mathbf{p}_k$ . In the bi-local CAT, the adaptation illuminant ( $\mathbf{p}^W$ ) is computed locally from the input image  $\mathbf{p}$ . This is the key difference between the bi-local CAT and both global and local CATs.

In fact, the bi-local CAT is a linear transformation of the guidance image  $\mathbf{g}$  in a window  $\omega_k$ , centered at pixel  $\mathbf{g}_k$  (*i.e.* admits the form of the linear transformation 11.8), for which:

$$\mathbf{a}_k^T = \begin{bmatrix} a_k^1 & 0 & 0 \\ 0 & a_k^2 & 0 \\ 0 & 0 & a_k^3 \end{bmatrix} \quad (11.13)$$

$$\mathbf{b}_k = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad (11.14)$$

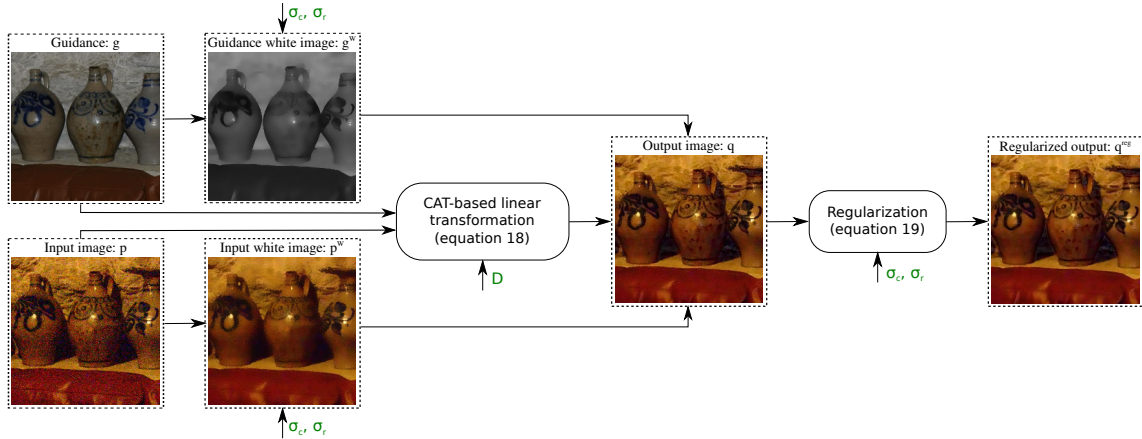
where

$$a_k^c = \frac{\mathbf{p}_k^W(c)}{\mathbf{g}_k^W(c)} D + (1 - D), \quad c \in \{1, 2, 3\}. \quad (11.15)$$

The bi-local CAT locally adapts the colors of the guidance  $\mathbf{g}$  to the colors of the input image  $\mathbf{p}$ . One of the key advantages of the bi-local CAT is that it is independent of the color distribution in each patch  $\omega_k$  as it does not rely on an assumption about the color distribution in  $\omega_k$ . This property ensures the robustness of the bi-local CAT in cases when two or more colors are present in a patch. Moreover, the bi-local CAT is built as a simple linear transformation of the guidance image  $\mathbf{g}$  and it is easy and fast to implement. We benefit from the aforementioned properties of the bi-local CAT by incorporating it into a novel CAT-based guidance filter, which is presented in the following section.

### 3.2 CAT-based guidance filter

We propose a new guidance-based filter, which carries out a linear transformation between the input image  $\mathbf{p}$  and the guidance  $\mathbf{g}$ . The input and guidance images can have a different number of channels. The output  $\mathbf{q}$  of our filter has the same number of channels as the input image. The flowchart of our method is illustrated in figure 11.2.



**Figure 11.2** – Flowchart of the CAT-based filter. The parameters of our method  $\sigma_c$ ,  $\sigma_r$  and  $D$  are given by the user and are displayed in green. To obtain the results in this paper, we use default values of the parameters (except in the figures comparing results with varying parameters).

Our filter admits the form of the linear transformation from equation 11.11. In the proposed filter, the solution for the matrix coefficient  $\mathbf{a}_k$  is derived using the bi-local CAT. We express  $\mathbf{a}_k$  as a diagonal matrix with diagonal elements  $a_k^c$ , computed with equation 11.15. That way, we compute each component  $\mathbf{q}_i^c$  of the output pixel  $\mathbf{q}_i$  as follows:

$$\mathbf{q}_i^c = a_k^c \underbrace{(\mathbf{g}_i^c - \mathbf{g}_k^W(c))}_{\text{high-pass / details}} + \underbrace{\mathbf{p}_k^W(c)}_{\text{low-pass}}, \text{ where} \quad (11.16)$$

$$a_k^c = \frac{\mathbf{p}_k^W(c)}{\mathbf{g}_k^W(c)} D + (1 - D). \quad (11.17)$$

Equation 11.16 shows that the outcome pixel  $\mathbf{q}_i$  is obtained by adding details from the guidance image to a low-pass version of the input image. The amount of the transferred details is controlled by  $a_k^c$ . The white images  $\mathbf{p}^W$  and  $\mathbf{q}^W$  are computed by applying the bilateral filter (as opposed to the averaging, used in the guided filter) with parameters  $\sigma_r$  and  $\sigma_c$ , which are the spatial and color deviations respectively. Each pixel  $\mathbf{p}_k^W$  ( $\mathbf{q}_k^W$ ) of the white image  $\mathbf{p}^W$  ( $\mathbf{g}^W$ ) is obtained as a weighted mean of the pixels in a window, centered at  $\mathbf{p}_k^W$  ( $\mathbf{q}_k^W$ ), i.e.  $\mu_{\mathbf{p}_k} = \mathbf{p}_k^W$  ( $\mu_{\mathbf{q}_k} = \mathbf{q}_k^W$ ).

As formula 11.17 is based on CAM, it is not an optimal solution to a minimization problem, unlike the coefficients in the guided filter and the Monge-Kantorovich mapping, which are computed during an optimization process. The linear regression in the guided filter is however sensitive to outliers, especially when two or more dominant colors are present in a patch. Moreover, the feasibility of the Monge-Kantorovich mapping depends

on the Gaussian assumption. In contrast, the CAT-based filter does not depend on a statistical model and it is distribution independent. The embedding of the bi-local CAT in the framework of our method ensures the robustness of the color adaptation in a patch. Our filter can be applied to a wide range of image processing tasks, as presented in section 5.

The spatial and color deviations  $\sigma_r$  and  $\sigma_c$ , used in the computation of the white images  $\mathbf{p}^W$  and  $\mathbf{q}^W$ , are the two parameters of the CAT-based filter. The parameters  $\sigma_r$  and  $\sigma_c$  are identical for both the input and guidance images, as the resolution of both images is the same. Figure 11.4 illustrates the influence of the two parameters on the output  $\mathbf{q}$ . The bigger the spatial deviation  $\sigma_r$ , the more details we recover from the guidance  $\mathbf{g}$  (details appear in the shadow areas with the increase of  $\sigma_r$ ). The detail recovery is also connected to the increase of the color deviation  $\sigma_c$ . The output  $\mathbf{q}$  gets sharper and less noisy with the increase of  $\sigma_c$ . The sharpness property is implicitly controlled by the ratio term  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c)$ . Figure 11.3 shows the influence of the ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c)$  on the detail recovery from the guidance image.

Like in the guided filter, the final value of the pixel  $\mathbf{q}_i$  is influenced by the linear transformations (equation 11.16) in all the windows  $\omega_k$ , covering  $\mathbf{q}_i$ . To obtain the final output pixel  $\mathbf{q}_i^{reg}$ , we regularize the pixel  $\mathbf{q}_i$ , by taking into account the values of  $\mathbf{q}_i$ , obtained in all windows  $\omega_k$ , containing  $\mathbf{q}_i$ :

$$\begin{aligned}\mathbf{q}_i^{reg} &= \frac{\sum_{k:i \in \omega_k} w_k (\mathbf{a}_k^T (\mathbf{g}_i - \mathbf{g}_k^W) + \mathbf{p}_k^W)}{\sum_{k:i \in \omega_k} w_k} \\ &= \hat{\mathbf{a}}_i^T (\mathbf{g}_i - \mathbf{g}_k^W) + \hat{\mathbf{p}}_i^W,\end{aligned}\quad (11.18)$$

where  $\hat{\mathbf{a}}_i^T = \frac{\sum_{k:i \in \omega_k} w_k \mathbf{a}_k^T}{\sum_{k:i \in \omega_k} w_k}$  and  $\hat{\mathbf{p}}_i^W = \frac{\sum_{k:i \in \omega_k} w_k \mathbf{p}_k^W}{\sum_{k:i \in \omega_k} w_k}$ . The scalar weights  $w_k$  are computed for each pixel  $\mathbf{q}_i$  using equation 11.2.

**Properties.** The properties of the CAT-based filter are presented hereafter:

**1. Robust detail recovery from the guidance image.** We show that we recover details from the guidance image by looking into several cases:

- The ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c) = 0$ , *i.e.* the window  $\omega_k$  is in an input shadow area. Then,  $\mathbf{q}_i = (1 - D)(\mathbf{g}_i - \mathbf{g}_k^W) + \mathbf{p}_k^W$  (from equation 11.16). In this case we have  $a_k^c = 1 - D > 0$  and we recover a certain amount of details from the guidance image (the amount depends on  $\sigma_c$  and  $\sigma_r$ ). This is not the case of the guided filter, as discussed in section 2.
- The ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c) \neq 0$ . Then,  $a_k^c > 0$  and we recover details from the guidance image.
- The ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c) = 1$ , *i.e.*  $\mathbf{p} = \mathbf{q}$ . Then,  $\mathbf{q}_i = \mathbf{g}_i$  and we recover the guidance image itself.

Our filter always recovers details from the guidance image, as  $\mathbf{p}_k^W/\mathbf{g}_k^W \geq 0 \geq (D - 1)/D$  for  $D \neq 0$  (which implies  $a_k^c \geq 0$  for all patches  $\omega_k$ , see equation 11.16). This makes the CAT-based filter suitable for texture transfer and detail enhancement, as it is presented in section 5.





**Figure 11.3** – Influence of the ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c)$  on the detail recovery from the guidance image. This influence is best observed when the guidance image is semantically different from the input image (*i.e.* when our method is used in the context of texture transfer). If we excluded the ratio  $\mathbf{p}_k^W(c)/\mathbf{g}_k^W(c)$  from the CAT-based filter (equation 11.16), we would recover less details (or no details for  $\sigma_c = 0.04 \times rg$ ) from the guidance image than if we applied the proposed CAT-based filter without any modifications. For the results, shown in this figure, the spatial deviation  $\sigma_r$  is fixed to  $diag/25$ , where  $diag$  denotes the image diagonal.

2. **Depth of field recovery.** Bigger values of  $\sigma_c$  result in a higher depth of field because  $\sigma_c$  controls the edge sharpness of the output image (figure 11.4). As a remark, small values of  $\sigma_c$  imply that  $\mathbf{g}_i - \mu_k \approx \vec{0}$ . Therefore, by using small values of  $\sigma_c$ , we obtain a result, similar to the input image (small values of  $\sigma_c$  have little/no effect when our method is applied for image denoising; see figure 11.4).
3. **Input light reflection recovery.** The CAT-based filter recovers light reflections (*e.g.* reflections of a light source, indirect object reflections, etc.) from the input image, as shown in figure 11.4 (and figure 11.5 in section 5). This is a result of the bi-local color adaptation, performed in a patch-wise manner. Our filter recovers input reflections even when used in the context of image denoising (unlike the guided filter, which exhibits a trade-off between the proper noise removal and the integrity

of the input light reflections; see figure 11.1). However, in the process of image denoising, we may compromise the input lighting atmosphere by introducing flash reflections from the guidance image, as shown in figure 11.4.

Our filter is usually carried out once with static input and guidance images. Our experiments show that for certain applications, such as image deblurring and image denoising, the CAT-based filter can be performed iteratively for a stronger effect. For such applications, we carry out our filter iteratively with a *dynamically changing input image*, as follows:

$$\mathbf{q}^{(t)} = \begin{cases} biCAT(\mathbf{g}, \mathbf{p}), & \text{if } t = 1; \\ biCAT(\mathbf{g}, \mathbf{q}^{(t-1)}), & \text{if } t \in [2, T]. \end{cases} \quad (11.19)$$

During each iteration  $t$ , the input image  $\mathbf{p}$  is updated and set to the result from the previous iteration  $\mathbf{q}^{(t-1)}$ . As a remark, the coefficient  $\mathbf{a}_k$  is recomputed during each iteration  $t$  using formula 11.17, where  $\mathbf{p}^W$  is replaced by the white image of the result  $\mathbf{q}^{(t-1)}$ . As a consequence of the use of a dynamic input image, we achieve a more robust and efficient transfer of details from the guidance image. As we always recover details from the guidance image (from property 1:  $a_k^c \geq 0$ ), each dynamic input image will be sharper and more detailed than the previous one.

For other applications, such as face beautification, the bi-local CAT is performed iteratively with a *dynamically changing guidance image*. In this case, we begin the iteration process with a smooth guidance image and during each iteration, we update the guidance image with the result from the previous iteration as follows:

$$\mathbf{q}^{(t)} = \begin{cases} biCAT(\mathbf{g}, \mathbf{p}), & \text{if } t = 1; \\ biCAT(\mathbf{q}^{(t-1)}, \mathbf{p}), & \text{if } t \in [2, T]. \end{cases} \quad (11.20)$$

For each application in section 5, we discuss which iteration method is used (if any) and the number of performed iterations.

## 4 Implementation

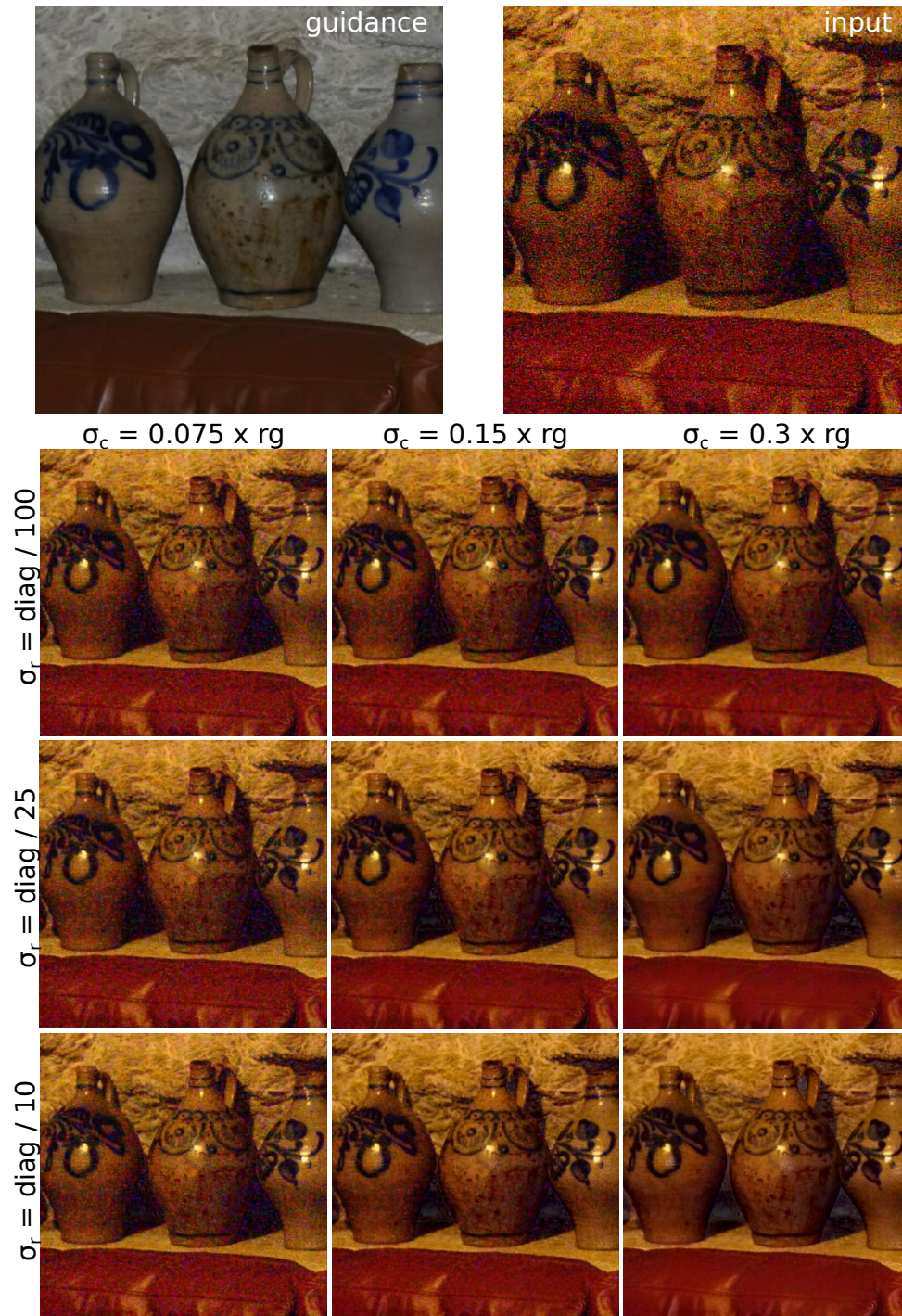
The implementation of our filter is presented as a pseudo code in algorithm 5. We first compute the white images of both the input and the guidance images using a bilateral filter. These white images are involved in the computation of the matrix coefficient  $\mathbf{a}_k$  (equation 11.17). In our implementation, the three diagonal elements  $a_k^c$  of the matrix  $\mathbf{a}_k^T$  are stored as pixel values of a new image  $\mathbf{a}$ . To perform the regularization from equation 11.18, we carry out a bilateral filter on the image  $\mathbf{a}$  and the white image  $\mathbf{p}^W$  respectively, and obtain images  $\hat{\mathbf{a}}$  and  $\hat{\mathbf{p}}^W$ . The images  $\hat{\mathbf{a}}$  and  $\hat{\mathbf{p}}^W$  contain the regularized pixels  $\hat{\mathbf{a}}_i$  and  $\hat{\mathbf{p}}_i^W$ , appearing in equation 11.18, and are used to compute the final output  $\mathbf{q}^{reg}$ .

## 5 Results

The proposed CAT-based filter is adopted for a number of applications. Hereafter, we present six main applications of our filter, namely image denoising, image texture trans-



fer, detail enhancement with NIR images, image deblurring, mask refinement and skin beautification.



**Figure 11.4** – Influence of the two parameters of the CAT-based filter, the spatial deviation  $\sigma_r$  and the color deviation  $\sigma_c$ . When increasing  $\sigma_c$ , we obtain a less noisy result. A bigger  $\sigma_c$  results in smoother white images and this benefits the denoising as well as the detail transfer. The increase of  $\sigma_r$  also influences the amount of details, which are transferred from the guidance image. More details (in the shadows, on the wall, on the pots) from the guidance image are recovered when increasing  $\sigma_r$ . Finally, some of the flash reflections are also transferred when  $\sigma_r$  gets bigger.



---

**Algorithm 5** CAT-based guidance filter
 

---

```

1: procedure CATFILTER
2: input:
3:    $\mathbf{p} \leftarrow (\mathbf{p}_1, \dots, \mathbf{p}_N)$  ▷ input image
4:    $\mathbf{g} \leftarrow (\mathbf{g}_1, \dots, \mathbf{g}_N)$  ▷ guidance image
5:    $i \leftarrow 0$  ▷ loop index
6: output:
7:    $\mathbf{a} \leftarrow (\mathbf{a}_1, \dots, \mathbf{a}_N)$  ▷ computed in loop
8:    $\mathbf{q}^{reg} \leftarrow (\mathbf{q}_1, \dots, \mathbf{q}_N)$  ▷ computed in regularization
9: while images: ▷ eqn. 11.2, where  $\mathbf{g}$  is replaced by  $\mathbf{p}$ 
10:   $\mathbf{p}^W = \text{BilateralFilter}(\mathbf{p}, \sigma_r, \sigma_c)$ 
11:   $\mathbf{g}^W = \text{BilateralFilter}(\mathbf{g}, \sigma_r, \sigma_c)$ 
12: loop:
13:   $\mathbf{a}_k \leftarrow \text{Compute-}a_k\text{-with-CAT}(\mathbf{p}_i^W, \mathbf{p}_i^W, D)$  ▷ eqn. 11.15
14:   $i \leftarrow i + 1$ 
15:  if  $i < N$  then
16:    goto loop
17:   $i \leftarrow 0$ 
18:   $\hat{\mathbf{p}}^W = \text{BilateralFilter}(\mathbf{p}^W, \sigma_r, \sigma_c)$  ▷ eqn. 11.2, where  $\mathbf{g}$  is replaced by  $\mathbf{p}$ 
19:   $\hat{\mathbf{a}} = \text{BilateralFilter}(\mathbf{a}, \sigma_r, \sigma_c)$  ▷ eqn. 11.2, where  $\mathbf{g}$  is replaced by  $\mathbf{p}$ 
20: regularization loop:
21:   $\mathbf{q}_i^{reg} = \text{PerformReg}(\mathbf{g}_i, \mathbf{g}_i^W, \hat{\mathbf{p}}_i^W, \hat{\mathbf{a}}_i)$  ▷ eqn. 11.18
22:   $i \leftarrow i + 1$ 
23:  if  $i < N$  then
24:    goto regularization loop
    
```

---

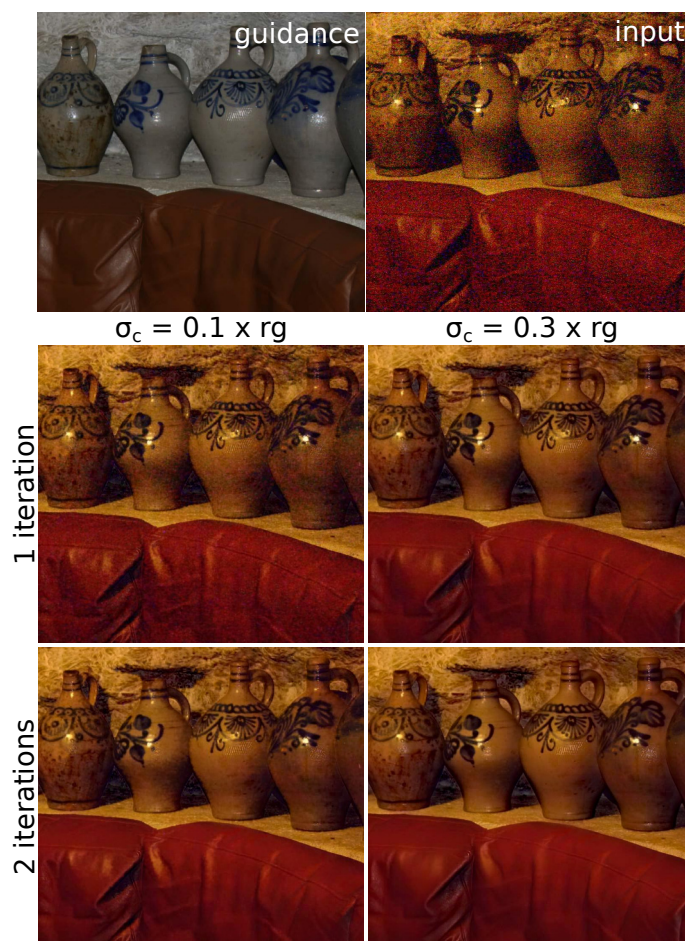
**Image denoising.** We adopt the CAT-based guidance filter in the context of example-based denoising. The input image  $\mathbf{p}$  is the noisy image (*e.g.* a non-flash image), whereas the guidance image  $\mathbf{g}$  is chosen to be a sharp noise-free image (*e.g.* a flash image). We aim to denoise image  $\mathbf{p}$  while maintaining its integrity (*i.e.* while preserving its ambient atmosphere in terms of input edges, reflections, etc.). Figure 11.5 shows a comparison between the CAT-based filter, the guided filter, the cross-bilateral filter and the patch-wise Monge-Kantorovich mapping, used in the context of image denoising. The guided and the cross-bilateral filters smooth down input edges and reflections, which results in a loss of sharpness and depth of field. In contrast, the patch-wise Monge-Kantorovich mapping increases the depth of the result but it also creates gradient reversal artifacts around the edges. The CAT-based filter removes noise while recovering details from the guidance image. Moreover, our filter preserves the integrity of the input lighting (the input reflections are kept in the result). Depending on the values of the parameters  $\sigma_r$  and  $\sigma_c$ , flash reflections from the guidance image may appear.

Furthermore, when performed iteratively, our filter removes noise more robustly. Figure 11.6 shows that in two iterations, the CAT-based filter removes a significant portion of the noise (this noise removal also depends on the parameter  $\sigma_r$ ).

**Texture transfer.** The CAT-based filter can be used to transfer a given texture pattern onto an input image for the purpose of image stylization. We refer to this process as texture transfer. The texture pattern is given in the guidance image. Our method successfully transfers the pattern onto the input image thanks to property 1 (*i.e.* detail recovery from the guidance image). Users can control the amount of transferred texture by setting up the parameter  $\sigma_c$ .



**Figure 11.5** – Image denoising. Comparison between the CAT-based filter, the guided filter, the cross-bilateral filter and the patch-wise Monge-Kantorovich mapping. Our filter denoises the input image while preserving the original light reflections. Moreover, it does not smooth the edges (as the guided filter), which results in detail enhancement and depth of field recovery. The Monge-Kantorovich mapping also recovers details from the guidance. However, it creates halo artifacts around the edges and it is not robust to noise.



**Figure 11.6** – Influence of the iterations on the output for two different values of the parameter  $\sigma_c$ . For a small  $\sigma_c$ , our method cannot filter all the noise from the input image in one iteration. After a second iteration, the CAT-based filter removes most of the remaining noise. For a big  $\sigma_c$ , the difference between the outputs from the first and second iterations is less appealing. This is because  $\sigma_c$  has a great impact on the image denoising and big values of  $\sigma_c$  help produce a noise-free result during the first iteration of the method. In such cases, the second iteration helps to produce sharper, more detailed results. In conclusion, for small values of  $\sigma_c$ , the CAT-based filter requires more iterations to denoise the input image than for big values of  $\sigma_c$ . In contrast, using small values of  $\sigma_c$  preserves better the input ambient lighting (as we introduce less flash reflections/shadows).

Figure 11.7 illustrates a texture transfer, carried out with the CAT-based filter as well as with the cross-bilateral and guided filters, and the screened Poisson equation [88, 92]. The screened Poisson equation is used for image sharpening and contrast enhancement. Given a target image gradient field (*e.g.* computed from a target image, which, in our case, is the texture pattern), the screened Poisson solver transfers it to an input image through an optimization process.



**Figure 11.7** – Texture transfer. Comparison between the CAT-based filter, the cross-bilateral filter, the guided filter and the Poisson solver. We use a binary mask and transfer the guidance texture pattern to the girl’s face. In one iteration, our method successfully recovers details and gradient information from the guidance image while maintaining the integrity of the input edges (we use  $\sigma_c = 30\%rg$ ). The Poisson solver needs 5 iterations to achieve a similar texture transfer to ours. The cross bilateral filter and the guided filter compromise the input edge information (they smoothen the input edges) and fail to transfer the texture information of the guidance image.

The CAT-based filter has a clear advantage over both cross-bilateral and guided filters, as it recovers details and gradient information from the guidance image, without

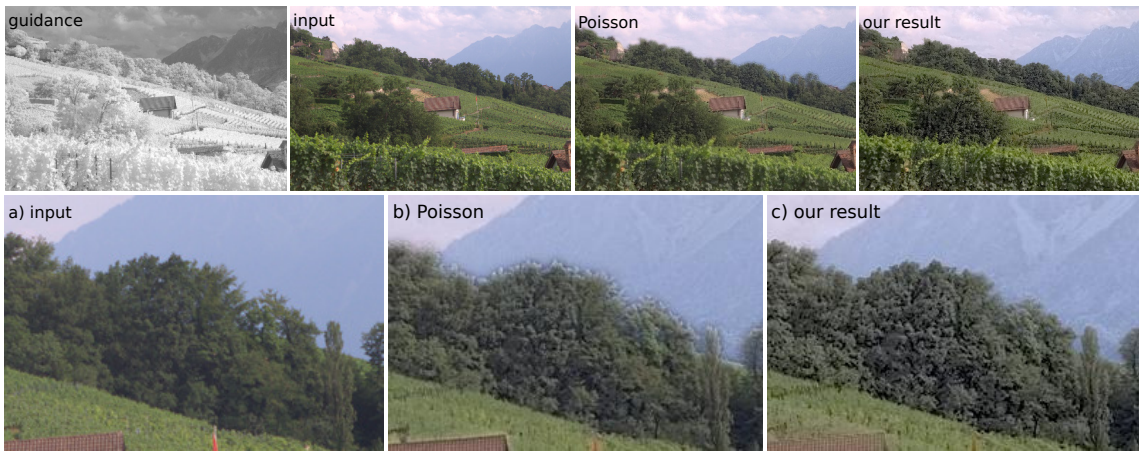


compromising the strong edges of the input image (figure 11.7). The CAT-based filter preserves only the gradient information from the guidance image. That way, the output of our texture transfer is a combination of gradient information and details from both the input image and the guidance.

The screened Poisson solver also succeeds to transfer texture from the guidance image and it creates a similar texture effect to that of our result. However, unlike our filter, the screened Poisson solver requires several iterations (5 for the example in figure 11.7). The amount of texture transfer in the screened Poisson equation is controlled by the number of iterations as well as by a parameter  $\lambda$ . To obtain the Poisson results in figure 11.7, we use  $\lambda = 0.2$ , *i.e.* we tolerate more texture transfer per iteration. The texture transfer in our method is controlled by the parameter  $\sigma_c$ , as shown in figure 11.3.

**Detail enhancement with NIR images.** NIR images contain more details and texture information than what standard digital cameras can capture. That is why, several methods [91, 142] adopt NIR images to enhance the details of RGB images. These methods rely on gradient optimization process, wavelet decompositions and texture transfer. In contrast, to carry out an image enhancement, we apply a single deterministic transformation.

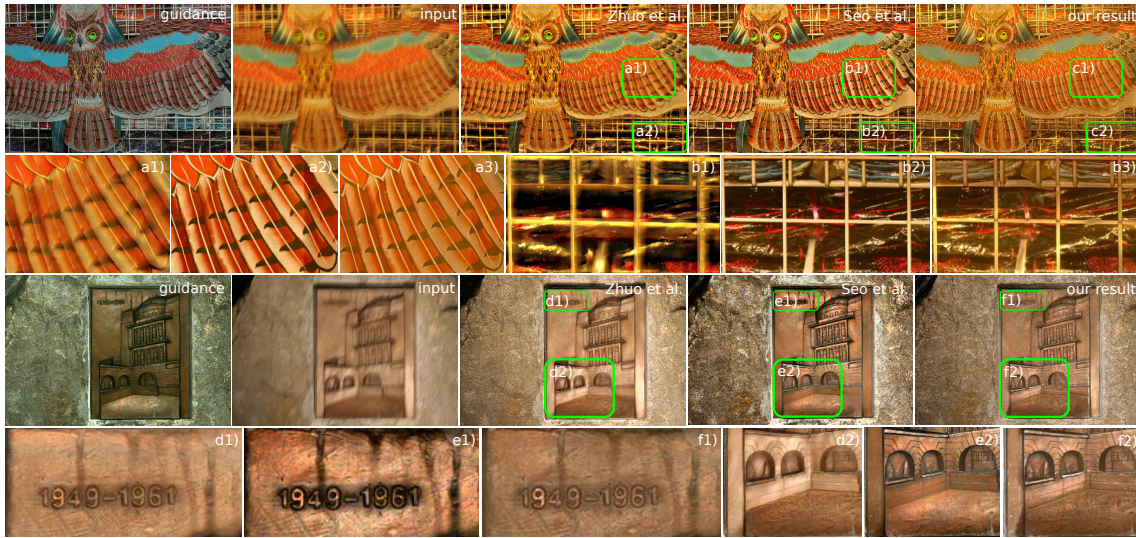
The CAT-based filter is applied between the guidance image  $\mathbf{g}$ , *i.e.* the NIR image, and the input image  $\mathbf{p}$ , *i.e.* the RGB image. Figure 11.8 shows our result for detail enhancement. We compare it with a result from the screened Poisson equation. As shown in figure 11.8, the screened Poisson solver sharpens some of the input edges but it also creates gradient reversal artifacts around them. Moreover, the Poisson solver fails to transfer the texture details from the NIR image. In contrast, the CAT-based filter enhances the input gradient information without causing visual artifacts and it transfers a lot of fine texture details from the NIR image.



**Figure 11.8** – Detail enhancement with an NIR image. The input image contains overexposed regions (see the sky and the mountain slope, snippet a). Our filter recovers the details in the input overexposed regions and enhances the texture in the rest of the image regions (our result appears sharper and more detailed than the input image, snippet c). Unlike our method, the screened Poisson solver [88] recovers less details and creates haloing artifacts around some of the edges (snippet b).

**Image deblurring.** We apply the CAT-based filter to recover a sharp image from a blurry one. Like the methods in [143–146], we use a guidance image in the deblurring

process. The guidance image can be a flash image, dark flash image, NIR image, etc. To recover a sharp image from the blurry input image, we iterate the CAT-based filter with a dynamically changing input image (as discussed in section 3). We obtain a sharp deblurred result after four iterations of the CAT-based filter, as shown in figure 11.9. Our method outperforms Zhuo et al.’s method [145] and performs fairly against Seo et al.’s method [144]. We obtain sharper and more detailed results than those of Zhuo et al. (Zhuo et al.’s results contain blur from the input image) and we preserve the input color distribution better than Seo et al. (as illustrated in figure 11.9). Seo et al. obtain very sharp results but they compromise the integrity of the input lighting (they transfer highlights from the guidance image, as shown in snippets e1 and e2 and explained in the caption of figure 11.9). In contrast, our results preserve the ambience of the input light source (snippet f2 in figure 11.9).

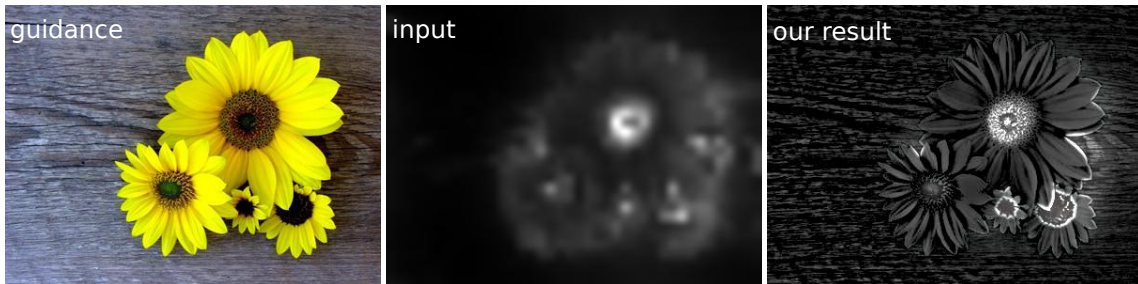


**Figure 11.9** – Iterative image deblurring. We carry out the bi-local CAT four times with a dynamic input image. We obtain sharp results, in which the blur from the input image is removed and the details from the guidance image are transferred. A comparison between the snippets a1 and c1 shows that our method outperforms Zhuo et al.’s method [145]. Unlike our result, Zhuo et al.’s result is significantly blurry (snippets a1 and d1). On the other hand, Seo et al.’s result [144] is visually sharper than both our result and Zhuo et al.’s result (snippets b1 and b2). However, Seo et al.’s method significantly changes the input color distribution (the colors appear more vivid and the overall contrast is increased). This is observed in snippets f2 and e2, where new highlight areas from the guidance image are introduced (flash highlights appear around the engraved digits in snippet e1 and on the floor in snippet e2). This compromises the integrity of the input lighting.

**Mask refinement.** Our filter is also applied for image mask refinement by using an RGB image as a guidance image. In this case, the input image is a mask: one-channel, binary or non-binary, computed from the RGB guidance image. This mask can be a foreground/background mask, a saliency map, etc. We aim to refine the input mask so that it represents the high-frequency content of the original RGB image. In practice, this application can be used for obtaining a more precise background/foreground mask or to recover a sharp detailed mask from a corrupted one. Figure 11.10 illustrates a mask refinement, using the saliency map of the guidance image. The chosen saliency map [101]



is blurry and lacks any texture details. In four iterations of the CAT-based filter (with a dynamic input image), we obtain a sharp detailed mask (figure 11.10).



**Figure 11.10** – Mask refinement. The input image is a blurry saliency map [101], computed from the guidance image. Our result is a sharp, detailed mask. The CAT-based filter was performed four times with a dynamic input image.

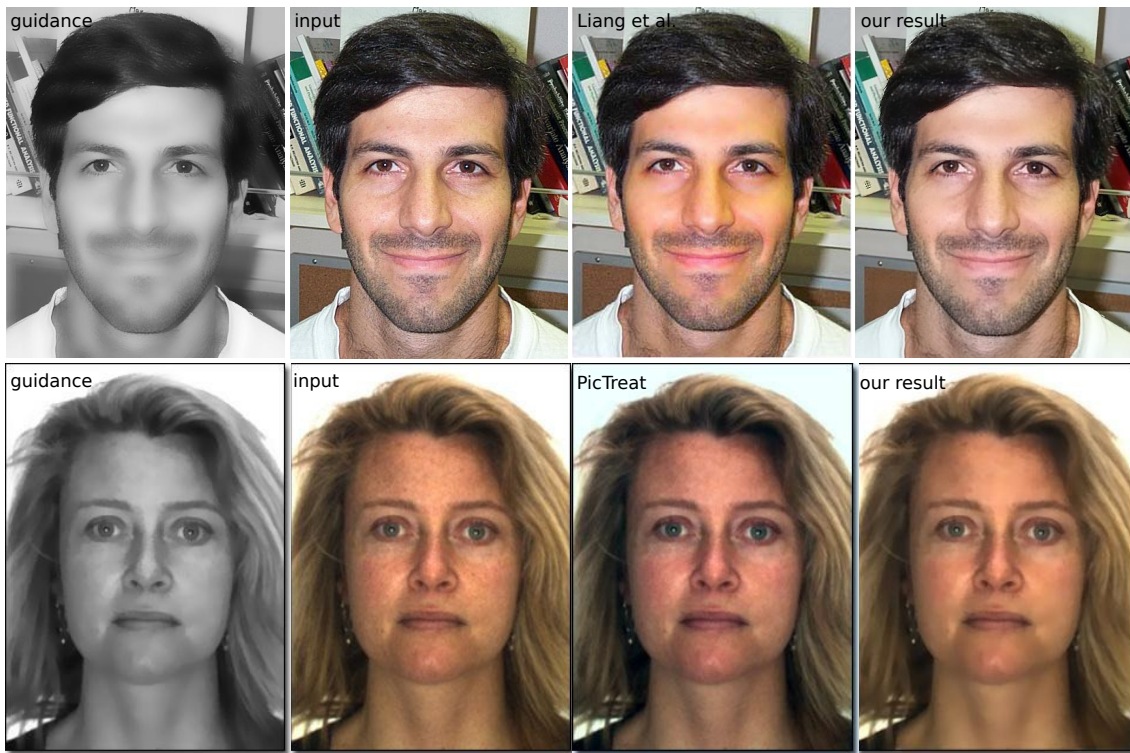
**Skin beautification.** We apply the CAT-based filter in the context of skin beautification. We aim to retouch an input face and to make it more pleasing by removing wrinkles, freckles, pigmentation, skin spots and glow, etc. We carry out two types of skin beautification - NIR skin beautification and single-image skin beautification.



**Figure 11.11** – Skin beautification with NIR image. Our result is obtained with one iteration of the CAT-based filter. We compare our result with a result from [147]. The smoothness of the skin in the NIR image is preserved in both results and input skin imperfections are removed. However, our result appears sharper than Susstrunck et al.’s result, because it additionally enhances the details from the NIR image (as discussed in the paragraph *detail enhancement with NIR images*).

1. **Skin beautification with NIR image.** This type of skin beautification uses NIR image as a guidance image. The NIR part of the spectrum is not very sensitive to skin “defects” and therefore, facial skin appears much smoother in NIR images [149]. Figure 11.11 shows a result for skin enhancement using our CAT-based filter. We smooth skin irregularities, such as wrinkles and spots, without compromising the overall sharpness of the output (on the contrary, we even increase the sharpness of the output). We compare our result with a result from the state-of-the-art method in [149].
2. **Single-image skin beautification.** We also perform a skin beautification using a single input image. First, we compute the guidance image directly from the input image by extracting its lightness. The input lightness corresponds to the lightness channel of the CIE Lab color space. The input lightness contains the skin “defects”,





**Figure 11.12** – Single-image skin beautification. Our CAT-based filter is carried out with a guidance image, corresponding to the filtered input lightness. We use our iterations of the CAT-based filter and a dynamic guidance image. Our filter successfully removes skin imperfections, wrinkles and freckles without compromising the sharpness of the input image. We obtain a result, similar to the result from Liang et al.’s method [148], without changing the facial color (first row). Furthermore, we remove a big part of the facial freckles (second row), which is not the case with the online application PicTreat (which also changes the input colors).

which we aim to remove. Therefore, we smooth the input lightness using bilateral filter, which preserves the edges of the input lightness. This filtered input lightness serves as a guidance image. Next, we perform the CAT-based filter iteratively with a dynamic guidance image in order to recover the input depth of field while preserving the skin smoothness. In figure 11.12, we compare our results with a result from Liang et al.’s skin enhancement method [148] and a result from the free online application PicTreat [150]. Our filter successfully removes skin freckles, smooths skin imperfections and reduces skin glow. Moreover, the CAT-based filter preserves the input color distribution, which is not the case with the result from PicTreat. Our results compare fairly to Liang et al.’s method. However, the latter may introduce unnatural skin color (figure 11.12). Moreover, Liang et al. use a skin mask, whereas we carry out our filter between entire images and do not require facial skin segmentation. The single-image skin beautification is a special case of single-image denoising. The described procedure can be applied to denoise an image without the use of a guidance image.

## **6 Conclusion**

In this paper, we have presented our CAT-based guidance filter. It consists of a patch-wise linear transformation between images. We have incorporated our novel bi-local CAT into the patch-wise linear transformation of our filter. That way, our CAT-based filter recovers details from the guidance image more robustly without compromising the input color distribution. We have applied our filter to a number of image processing applications, such as image denoising, image deblurring, texture transfer, etc. Furthermore, our results have shown the efficiency of our method in terms of sharpness and depth of field recovery, detail enhancement and facial skin beautification. The key advantage of our CAT-based filter over existing guidance-based filters lies in its simplicity and its applicability to various image processing domains.



---

## 1 Conclusion

In this thesis, we have focused on image editing techniques for stylizing user photos and for improving their quality. More specifically, we have contributed to the field of example-guided image editing. We have addressed open questions from three main topics in this field, *i.e.* color transfer, HDR imaging and guidance-based image filtering.

First, we have explored how to stylize images using a color transfer. Similarly to many state-of-the-art methods, we have addressed color transfer as a problem of transforming statistical distributions. To this end, we have exploited the potential of three distribution models, *i.e.* GMMs, MGGD and Beta distribution models, in the context of color transfer. To overcome main limitations of existing color transfer methods, we have proposed a local color transfer method, based on GMMs. Incorporating four mapping policies into our method has ensured its robustness for image editing tasks, such as photo-realistic stylization. Our mapping policies have explained key photographic approaches, in which light and color are adjusted to create artistic effects. Apart from color and light, other image features, such as image gradient, also impact the appearance of an image. To extend the color transfer to a feature transfer between images, we have proposed an MGGD-based transformation. We have applied the proposed transformation to carry out a simultaneous color and gradient transfer between input and target images. The joint transfer of color and gradient have proven beneficial for representing the look of the target image.

The distributions of image features, though discrete, have commonly been modelled using continuous distributions (mostly using the MGD). To address this discrepancy, we have adopted bounded distributions. We have experimentally found out that the bounded Beta distribution accurately fits most color and light distributions in images and accounts for their skewness. To exploit this property of the Beta distribution, we have presented a new color transfer method, for which the color and light distributions in images are modelled using the Beta distribution. To this end, we have introduced a novel transformation of the Beta distribution. Compared with recent state-of-the-art methods, the results, obtained with our Beta-based transformation, appeared more natural and less saturated.

The comparison between results from various color transfer methods is usually based solely on human perception and thus, it is subjective. In this thesis, we aimed to lessen the level of subjectivity when evaluating the quality of a color transfer. To achieve this, we have proposed an objective metric which explains the relationship between human perception and a number of image features.

Furthermore, this thesis focused also on HDR imaging. First, we have extended existing LDR color transfer methods to HDR images. We have shown that the proposed extension is required in order to ensure the plausibility of the final results. Second, we

have developed a new method for creating HDR images. Nowadays, HDR images are commonly created using a sequence of (in general, more than two) multi-exposure images. To create artifact-free HDR images, users are required to use a tripod and to carefully adjust the exposure time. Therefore, the shooting process may be time-consuming. Moreover, multi-exposure methods may cause ghosting artifacts for scenes with moving objects. To tackle the drawbacks of multi-exposure methods, we have introduced a new method which uses only two LDR images, *i.e.* flash and non-flash images, to reconstruct the dynamic range of real-world scenes. To achieve this, we have mimicked the camera response function by a brightness function to obtain a number of differently exposed images from the non-flash image. We have then used our new chromatic adaptation transform (CAT), called bi-local CAT, to recover scene details from the flash image. Our experiments have shown that our method (using only two LDR images) compares well with classical multi-exposure methods (using more than two multi-exposure images).

In the context of the HDR image creation, the bi-local CAT recovers details from the flash image, removes flash shadows and reflections. We have gone even further and we have exploited the potential of the bi-local CAT for various image editing applications such as image de-noising, image de-blurring, texture transfer, etc. We have proposed a novel guidance-based filter in which we have incorporated the bi-local CAT. We have shown that the proposed filter performs as good as (and for certain applications even better than) state-of-the-art methods.

---

## 2 Future work

The works, presented in this thesis, may provide answers to a number of research questions, but they also open new avenues for improvement. Hereafter, we discuss future work.

---

### 2.1 Example-guided color transfer

Most color transfer methods (including the ones proposed in this thesis) require a target image as an input. The target image is provided by the user and it reflects the intention of the user to stylize an input image in a certain way. The target image plays a key role in the color transfer model. The features, which are extracted from the target image, are incorporated in the transformation, applied on the input image. This means that the color transfer methods focus on the specifics of the given target image rather than on the class (or style) of images that the target image belongs to (*e.g.* low-key/high-key images; images, characterized by more than one dominant color, low/high depth of field, etc). In this sense, color transfer is an ill-posed problem. Given two visually similar target images, most color transfer methods would produce two different results for the same input image. Providing a single target image makes it difficult to predict the type of modifications a user desires.

To better specify the color transfer problem and to provide greater user control over the colorizing process, a recent method replaces the target image with a target color palette [151]. In this case, the color palette is extracted from the colors of the input image. Each color in the color palette can then easily be modified in order to edit the input

image as desired. This allows the users greater creative freedom. Another recent method aims to guide the color transfer by proposing meaningful color changes with regard to the input content [152]. This method uses an internet database to learn color manifolds and to provide the user with the most appropriate color palette for a given object (*e.g.* the color palette of a banana ranges from green to yellow).

The aforementioned methods focus mainly on the color representation of the input image. Apart from color, the users may intent to edit other image features, such as image composition, depth of field, texture, etc. The combinations of those features define different image classes (styles). The human visual system distinguishes between any two types of images (or on the contrary, recognizes them as similar) by considering an ensemble of all the image features (and not analysing them separately). We believe that we could simulate this behaviour of the human perception using abstract images which represent various image styles in a low-level manner (regardless of the image content specifics and details). These abstract images could be incorporated in the framework of color transfer methods, replacing the target image. Such an approach would allow users to stylize their images quickly and efficiently and it will be tackled in the near future.

---

## 2.2 Multivariate Beta distribution

As shown in this thesis (chapter 7), the Beta distribution accurately describes light and color distributions of images. To this end, we have proposed a transformation of the Beta distribution, applied in the context of color transfer. The Beta distribution is a univariate distribution. Therefore, our Beta transformation is introduced as a 1D transformation, applied separately on each channel of the input image (considering the distribution features of the corresponding target channel). That is why, our transformation does not take into account the correlation between the channels of the color space. To overcome this limitation, we perform our Beta transformation in the CIE Lab color space, whose channels are close to independent.

The multivariate generalization (for more than two shape parameters) of the Beta distribution is the Dirichlet distribution [153]. The Dirichlet distribution is not as well studied as the Beta distribution. However, it has proven to be beneficial for retrieving similar images from a database and for skin color detection [154]. The Dirichlet distribution with four shape parameters could be adopted to fit any three-dimensional data, such as the joint distribution of color and light in images (by considering the correlation between color and light channels). Moreover, the multi-dimensional Dirichlet model could describe the joint distribution of multiple image features, such as gradient, saliency, texture descriptors, etc. Given the promising color transfer results, obtained using our Beta transformation, we believe that the Dirichlet distribution could be beneficial to image editing applications, such as color transfer and multi-dimensional feature transfer between images.

---

## 2.3 Fisher distribution

The Fisher distribution is a well-studied semi-bounded distribution. It is mainly used in the statistical analysis for hypothesis testing [155]. In this thesis, we have incorporated the Fisher distribution in the intermediate steps of our Beta transformation. We reckon that the Fisher distribution has a great potential for image processing tasks. For



instance, we believe that the absolute luminance of HDR images could be modelled by the Fisher distribution (or a mixture of Fisher distributions). Indeed, the luminance distribution shares common characteristics with the Fisher distribution, *e.g.* it is continuous and semi-bounded. Future experiments and analysis will answer the question whether the Fisher distribution could be beneficial to tone-mapping or inverse tone-mapping techniques.

---

## 2.4 Video color transfer

Color transfer can be extended to video sequences, in which case the color palette of a target video can automatically be transferred to a given input video. Applying naive color transfer between two video sequences may lead to artifacts. To improve the naive video color transfer, several example-guided video editing methods have been proposed [6, 44].

In this thesis, we have tackled color transfer between still images. In many ways, our color transfer methods outperform state-of-the-art color transfer methods, *e.g.* in terms of photo-realism, contrast preservation, feature transfer (we can perform multi-feature transfer using our MGGD-based transformation), etc. Therefore, in the future, we could exploit the potential of the proposed methods for video applications. The main challenge of extending our color transfer methods to videos would consist in preserving the coherence of the video frames over time, which would require including additional constraints to the color transfer process.

---

## 2.5 Style transfer

Recently, several methods for artistic style transfer have been proposed [52, 54, 55] (they are discussed in more details in the first chapter of part II). Most of those methods are based on convolutional neural networks which are trained to recognize image features at various image scales. This ensures a robust transfer of image features from a target image to an input image. Often, the target image is a painting and the deep learning methods are used to transfer the artist's style to the input image. Most style transfer methods, based on neural networks, produce impressive results for a target image which is a painting but are less suitable for a photo-realistic style transfer. Another main limitation of the deep learning methods is the significantly long time it takes to generate the final result.

In this thesis, we have introduced a method for photo-realistic style transfer. Our method determines the key style features of both the input and target images and incorporates them into the style transfer to obtain a result whose style is similar to the target style in terms of color and light. Our approach can be performed in real time between any two images (regardless of their content), which is an advantage over the deep learning methods for style transfer. However, there is room for improvements in order to produce equally impressive results to those of deep learning methods. For instance, even when performing a photo-realistic style transfer, texture patterns (such as skin, wood, water patterns, etc.), may appear and play a significant role in defining the target style. In this sense, it is important to increase the number of features which are considered during the style transfer. In this thesis, we have introduced an MGGD-based model for simultaneous transfer of color and gradient and we have shown the improvement over existing color transformations. Our ultimate goal for the future is to model style in images using a number of descriptive

image features. Such a model could be beneficial for extracting the key style features of a target image and transferring them properly to an input image. One of the ways this could be achieved is by extending our MGGD-based transformation using a mixture of MGGDs, and designing proper mapping strategies to map the different clusters of image features.

---

## 2.6 360-degrees images

Nowadays, the 360-degree images are becoming more and more popular. These panoramic spherical images allow users to visualize a scene in each possible direction as opposed to the flat two-dimensional images. The 360-degree images can easily be captured using applications, such as Google street view. Moreover, social platforms, such as Facebook, have begun supporting the upload and visualization of the panoramic images. These key factors have helped to increase the interest towards the 360-degree images.

The popularity of the 360-degree images has opened the question of designing image editing algorithms for such type of images. Existing image editing methods cannot be directly applied to the panoramic images. The 360-degree images need to be first projected onto a cube for retrieving six two-dimensional images. Image editing techniques can then be applied to the so-obtained images. The main challenge consists in preserving the integrity of the 360-degree image once the six edited images are stiched together to recover the panoramic image. For the future, we are interested in tackling image stylizing techniques, such as color and style transfers, image hallucination [156], etc., for 360-degree images.

---

## 2.7 HDR image creation

In this thesis, we have introduced a method for HDR creation from two images, flash and non-flash images. Our method is suitable for indoor scenes and dark environment scenes (with great dynamics) for which the reach of the flash is significant. For scenes in which the contribution of the flash is insignificant, such as night-time outdoor scenes, our method recovers only the details present in the flash image, *i.e.* the details highlighted by the flash. In this case, one idea for future improvement would be to replace the flash image by an image which brings more information about the scene, *e.g.* NIR image.

Using two images to create an HDR image is an improvement over classical multi-exposure methods. However, one way to go even further is to consider a single image. As previously discussed in chapter 10, Li et al. [125] have recently proposed an image brightening method, which fuses three differently exposed images, virtually created from a single image. However, Li et al.'s method has not been used to recover HDR images and therefore, it has not been compared against classical multi-exposure methods. Recovering the dynamic range of real-world scenes from a single image imposes a strong restriction. To recover an HDR image, we first need to compute the CRF and then, we need to recover the missing details in the under-/over-exposed image pixels. As shown in chapter 10, we can now mimic the CRF using a brightness function. The challenge now is how to extract enough information from the given image in order to truthfully represent the scene details. The possibility of using a single image for creating HDR images will be explored in the future.

## 2.8 HDR video sequences

HDR video sequences have successfully been created using an extension of the multi-exposure approach. Instead of capturing several multi-exposure images for each frame in the video, recent methods capture a video by alternating short and long exposures between the frames [157–159]. Then, the missing exposure for each frame is reconstructed from the neighbouring frames using motion estimation. Finally, the reconstructed exposures per frame are merged together and the HDR video sequence is created.

We believe that HDR video sequences can be generated using a similar approach. Instead of alternating short and long exposures, we can alternate non-flash and flash frames. That way, once we estimate the motion flow between two consequent flash and non-flash frames, we can recover an HDR image from those two frames using our method, proposed in chapter 10. This idea brings up two main challenges. First, to estimate the motion between flash and non-flash frames, we would need to account for the different illumination conditions of both frames. Second, and more important, a careful set-up would be required to shoot the flash/non-flash alternating video sequence. One such set-up has already been proposed by Hudon et al. [160]. Their set-up uses a Kinect depth sensor, a camera and a flash LED light, and generates flash/non-flash images pairs at 30 frames per second. We have already conducted several experiments using Hudon et al.’s experimental set-up and the results have proven promising.

# Appendices

# 13

## Appendix

### A Demonstration of the choice of $\hat{\tau}_{\mathbf{w}_i}$ and $\hat{\mathbf{e}}_{\mathbf{w}_i}$ in equations 6.12 and 6.13

First, for each  $\mathbf{w} \sim MGGD(\mathbf{M}_V, m_V, \beta_V)$ , formulas 6.12 and 6.13 are correctly formulated thanks to the positive definiteness of the matrix  $\mathbf{M}_V$ . As a result of the positive definiteness of  $\mathbf{M}_V$ , the inverse of the matrix  $\mathbf{M}_V$  and its square root are well-defined positive definite matrices.

The proof will be carried out in two steps. As a first step, we show that the vectors  $\hat{\tau}_{\mathbf{w}_i}$  and  $\hat{\mathbf{e}}_{\mathbf{w}_i}$ , as samples of the random variables  $\hat{\tau}_{\mathbf{w}}$  and  $\hat{\mathbf{e}}_{\mathbf{w}}$ , turn the equality in distribution 6.10 into a strict equality in values as follows:

$$\begin{aligned} \hat{\tau}_{\mathbf{w}_i} (m_V \mathbf{M}_V)^{\frac{1}{2}} \hat{\mathbf{e}}_{\mathbf{w}_i} &= \left\| (m_V \mathbf{M}_V)^{-\frac{1}{2}} \mathbf{w}_i \right\| (m_V \mathbf{M}_V)^{\frac{1}{2}} \\ &\quad \times \frac{(m_V \mathbf{M}_V)^{-\frac{1}{2}} \mathbf{w}_i}{\left\| (m_V \mathbf{M}_V)^{-\frac{1}{2}} \mathbf{w}_i \right\|} = \mathbf{w}_i. \end{aligned} \quad (\text{A1})$$

Secondly, we prove that the random variables  $\hat{\tau}_{\mathbf{w}}$  and  $\hat{\mathbf{e}}_{\mathbf{w}}$  have the same properties as  $\tau_{\mathbf{w}}$  and  $\mathbf{e}_{\mathbf{w}}$  (from the stochastic representation 6.10).

To begin with, after replacing the variable  $\mathbf{w}$  with its stochastic representation 6.10, the distribution of the random variable  $\hat{\tau}_{\mathbf{w}}^{2\beta_V}$  admits the following form:

$$\begin{aligned} \hat{\tau}_{\mathbf{w}}^{2\beta_V} &= \left\| (m_V \mathbf{M}_V)^{-\frac{1}{2}} \mathbf{w} \right\|^{2\beta_V} = \left( \mathbf{w}^T (m_V \mathbf{M}_V)^{-1} \mathbf{w} \right)^{\beta_V} \\ &\stackrel{d}{=} \left( \tau_{\mathbf{w}} (m_V \mathbf{M}_V)^{\frac{1}{2}} \mathbf{e}_{\mathbf{w}}^T (m_V \mathbf{M}_V)^{-1} \tau_{\mathbf{w}} (m_V \mathbf{M}_V)^{\frac{1}{2}} \mathbf{e}_{\mathbf{w}} \right)^{\beta_V} \\ &= \tau_{\mathbf{w}}^{2\beta_V} \left( \mathbf{e}_{\mathbf{w}}^T \mathbf{e}_{\mathbf{w}} \right)^{\beta_V} = \tau_{\mathbf{w}}^{2\beta_V} \|\mathbf{e}_{\mathbf{w}}\|^{2\beta_V} = \tau_{\mathbf{w}}^{2\beta_V}. \end{aligned} \quad (\text{A2})$$

Consequently, the random variable  $\hat{\tau}_{\mathbf{w}}$  is distributed similarly to  $\tau_{\mathbf{w}}$  (the distribution of which is presented in equation 6.7).

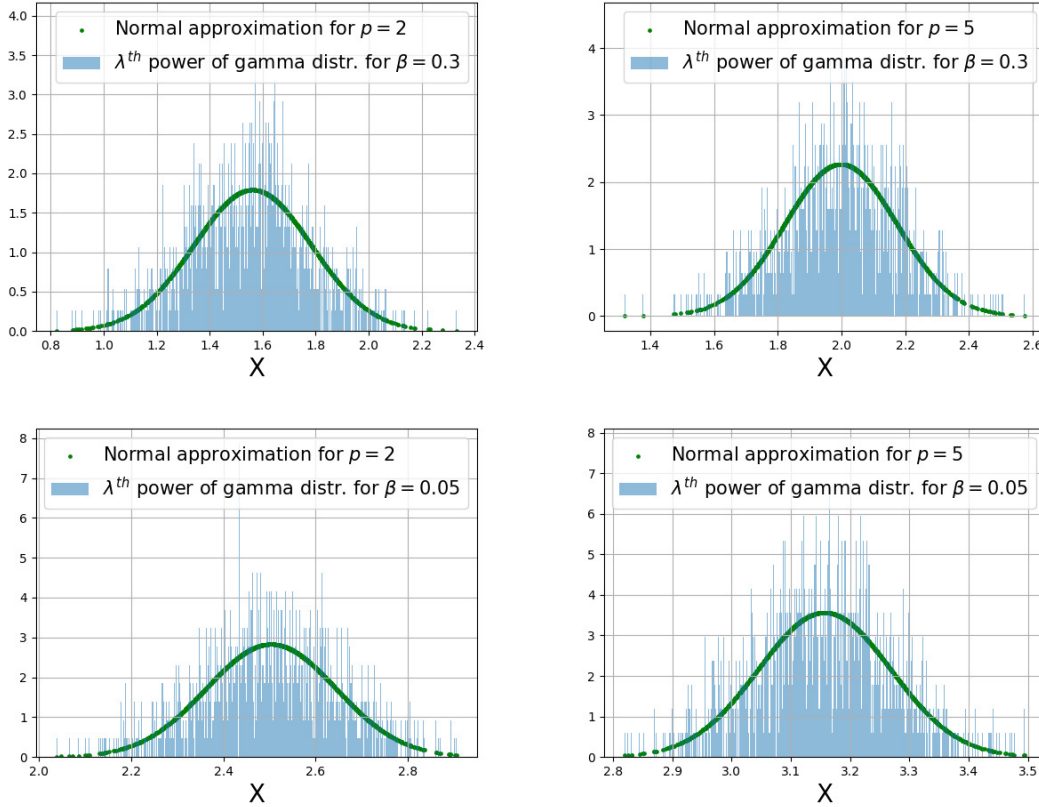
To conclude the proof, we show that the random variable  $\hat{\mathbf{e}}_{\mathbf{w}}$  is uniformly distributed on a unit sphere. By definition [161], the random variable  $\hat{\mathbf{e}}_{\mathbf{w}}$  is uniformly distributed on a unit sphere if (i)  $\|\hat{\mathbf{e}}_{\mathbf{w}_i}\| = 1$  and (ii) the distribution of  $\hat{\mathbf{e}}_{\mathbf{w}}$  is rotation invariant. It can be easily shown that  $\forall i \in \{1, \dots, n\}$  condition (i) is satisfied. To this end, we only need to prove that the distribution of the random variable  $\hat{\mathbf{e}}_{\mathbf{w}}$  is rotation invariant. As by definition (equation 6.12)  $\hat{\mathbf{e}}_{\mathbf{w}_i}$  is defined only in terms of the sample vector  $\mathbf{w}_i$  and its

Euclidean norm, the distribution of  $\hat{\mathbf{e}}_{\mathbf{w}}$  is rotation invariant if and only if the variable  $\mathbf{w}$  does not change under rotations. Thanks to the positive definitiveness of the matrix  $\mathbf{M}_{\mathbf{V}}$ , the density function of  $\mathbf{w}$  can be expressed as follows:

$$f(\mathbf{w}) = \frac{\gamma(\frac{p}{2})}{\pi^{\frac{p}{2}}} h_{\Sigma_{\mathbf{V}}, \beta_{\mathbf{V}}}(\mathbf{w}), \text{ where} \quad (\text{A3})$$

$$h_{\Sigma_{\mathbf{V}}, \beta_{\mathbf{V}}}(\mathbf{y}) = \frac{\beta_{\mathbf{U}}}{|\Sigma_{\mathbf{V}}|^{\frac{1}{2}} \gamma(\frac{p}{2\beta_{\mathbf{U}}})} \exp\left(-\frac{\|(m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{y}\|^2}{2m_{\mathbf{V}}^{\beta_{\mathbf{U}}}}\right). \quad (\text{A4})$$

As the Euclidean norm  $\|(m_{\mathbf{V}} \mathbf{M}_{\mathbf{V}})^{-\frac{1}{2}} \mathbf{w}\|$  does not change under rotations, the density function A3 is rotation invariant. Consequently  $\mathbf{w}$  does not change under rotations and therefore, neither does the random variable  $\hat{\mathbf{e}}_{\mathbf{w}}$ . Finally, as both conditions (i) and (ii) are satisfied,  $\hat{\mathbf{e}}_{\mathbf{w}}$  is randomly distributed on a unit sphere.

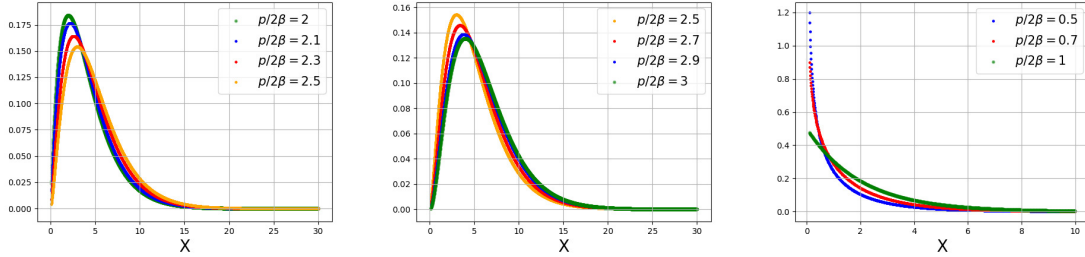


**Figure A1** – The plots illustrate the distributions of the  $\lambda^{th}$  power ( $\lambda = 1/4$ ) of samples, drawn from gamma distributions, and their corresponding normal approximations (from (B2) and (B3)). The sample distributions and their normal approximations are shown for different values of the dimension  $p$  and the shape parameter  $\beta$  of the gamma distribution.

## B Proof of Proposition 3.2.1

We first show that the distributions of  $\hat{\gamma}_{\mathbf{w}}^{2\lambda\beta_{\mathbf{U}}}$  and  $\hat{\gamma}_{\mathbf{g}}^{2\lambda\beta_{\mathbf{V}}}$  can be approximated by normal distributions. Then, we apply the Central limit theorem (CLT) to derive transformation

(6.16).



**Figure B1** – The plots illustrate gamma distributions with varying shape parameters and fixed scale parameters equal to 2 (from assumptions (6.18) and (6.19)). A gamma distribution  $\Gamma(\beta, \alpha)$  with a non-integer shape parameter  $\beta$  can be approximated by the gamma distribution  $\Gamma([\beta], \alpha)$  (the green curves). As illustrated by the plots, the approximation error is not significant (except for shape parameters less than 1), which shows the quality of our assumptions. When the shape parameter of the gamma distribution is less than 1, *i.e.*  $p = 1$ , approximations (6.18) and (6.19) are less accurate.

The shape parameters  $\beta_U$  and  $\beta_V$  in equations (6.14) and (6.15) take values in the range  $(0, 1]$ . Therefore, the following inequalities are satisfied:

$$p_U := \frac{p}{2\beta_U} \geq 1 \text{ and } p_V := \frac{p}{2\beta_V} \geq 1 \forall p \geq 2, \quad (\text{B1})$$

where  $p \in \mathbb{N}$  denotes the dimension. When  $p_U$  and  $p_V$  are large enough (*i.e.*  $p_U \mapsto \infty$  and  $p_V \mapsto \infty$ ), the distributions  $\Gamma(p_U, 2)$  and  $\Gamma(p_V, 2)$  of  $\hat{\tau}_w^{2\lambda\beta_U}$  and  $\hat{\tau}_g^{2\lambda\beta_V}$  can accurately be approximated by normal distributions. For  $p_U$  and  $p_V$  to be large,  $\beta_U$  and  $\beta_V$  need to be small, *i.e.*  $\beta_U \ll 1$  and  $\beta_V \ll 1$ . However, we aim to build a general transformation, which is valid for all shape parameters  $\beta_U$  and  $\beta_V$  (including those, close to 1). To this end, instead of directly approximating  $\Gamma(p_U, 2)$  and  $\Gamma(p_V, 2)$  by normal distributions, we approximate their  $\lambda^{th}$  power by normal distributions [162]:

$$\Gamma^\lambda(p_U, 2) \sim \mathbf{N}(\mu_\lambda, \sigma_\lambda^2), \quad (\text{B2})$$

$$\Gamma^\lambda(p_V, 2) \sim \mathbf{N}(\mu'_\lambda, \sigma'^2_\lambda), \quad (\text{B3})$$

where  $\mu_\lambda$ ,  $\mu'_\lambda$ ,  $\sigma_\lambda^2$  and  $\sigma'^2_\lambda$  are defined in equation (6.17) (as discussed in [85]). Although the  $\lambda^{th}$  power of a gamma distribution converges toward a normal distribution faster than the gamma distribution itself, the shape parameters  $p_U$  and  $p_V$  still need to be large enough for a good approximation. To improve approximations (B2) and (B3) for small values of  $p_U$  and  $p_V$ , we set  $\lambda$  to  $1/4$ , as proposed by Hawkins et al. [98]. To demonstrate the quality of approximations (B2) and (B3), we compare the  $\lambda^{th}$  power of the gamma distribution to its corresponding normal approximation for two different values of the shape parameter  $\beta$  and the dimension  $p$ . Figure A1 shows that the normal distribution (with parameters defined in (6.17)) fits very accurately the distribution of the  $\lambda^{th}$  power of samples, drawn from a gamma distribution.

In general,  $p_U$  and  $p_V$  are not integers. As approximations (B2) and (B3) are valid for integer values  $p_U$  and  $p_V$ , they can be carried out only under assumptions (6.18) and (6.19) of proposition 3.2.1. We demonstrate the quality of the assumed approximations (6.18) and (6.19) in figure B1.



So far, we have shown that  $\hat{\tau}_{\mathbf{w}}^{2\lambda\beta_{\mathbf{w}}} \sim \mathbf{N}(\mu_{\lambda}, \sigma_{\lambda}^2)$  and  $\hat{\tau}_{\mathbf{g}}^{2\lambda\beta_{\mathbf{g}}} \sim \mathbf{N}(\mu'_{\lambda}, \sigma_{\lambda}'^2)$ . Now, we apply the CLT between  $\hat{\tau}_{\mathbf{w}}^{2\lambda\beta_{\mathbf{w}}}$  and  $\hat{\tau}_{\mathbf{g}}^{2\lambda\beta_{\mathbf{g}}}$ , and compute the sample vectors  $\hat{\tau}'_i$  of a new variable  $\hat{\tau}'$ :

$$\hat{\tau}' = \frac{(\hat{\tau}_{\mathbf{w}_i})^{2\lambda\beta_{\mathbf{w}}} - \mu_{\lambda}}{\sigma_{\lambda}} \sigma'_{\lambda} + \mu'_{\lambda}, \quad (\text{B4})$$

The variable  $\hat{\tau}'$  is approximately distributed as  $\mathbf{N}(\mu'_{\lambda}, \sigma_{\lambda}'^2)$  and from approximation (B3) it follows:

$$(\tau')^{\frac{2\beta_{\mathbf{v}}}{\lambda}} \sim \Gamma\left(\frac{p}{2\beta_{\mathbf{v}}}, 2\right), \quad (\text{B5})$$

*i.e.*  $(\hat{\tau}')^{\frac{2\beta_{\mathbf{v}}}{\lambda}} \sim \hat{\tau}_{\mathbf{g}}^{2\beta_{\mathbf{v}}}$ . Now, transformation (6.16) is directly derived from transformation (B4).



# Bibliography

- [1] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, A. Trémeau, E. Reinhard *et al.*, “A survey of color mapping and its applications.” in *Eurographics (State of the Art Reports)*, 2014, pp. 43–67. [18](#), [44](#)
- [2] M. Oliveira, A. D. Sappa, and V. Santos, “Unsupervised local color correction for coarsely registered images,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 201–208. [18](#), [66](#)
- [3] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001. [18](#), [41](#), [45](#), [58](#), [61](#), [62](#), [66](#), [82](#), [85](#), [99](#), [100](#), [104](#), [111](#), [120](#), [121](#), [149](#)
- [4] Z. Ma and A. Leijon, “Beta mixture models and the application to image classification,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 2045–2048. [18](#), [38](#), [91](#)
- [5] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentation by expectation-maximization,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 747–754. [18](#), [36](#), [46](#), [58](#), [61](#), [62](#), [66](#), [85](#), [112](#), [120](#), [121](#)
- [6] N. Bonneel, K. Sunkavalli, S. Paris, and H. Pfister, “Example-based video color grading,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 39–1, 2013. [18](#), [41](#), [46](#), [51](#), [53](#), [54](#), [56](#), [58](#), [61](#), [62](#), [66](#), [85](#), [92](#), [99](#), [100](#), [101](#), [104](#), [112](#), [113](#), [114](#), [120](#), [121](#), [149](#), [168](#)
- [7] H. Hristova, O. Le Meur, R. Cozot, and K. Bouatouch, “Style-aware robust color transfer,” in *Proceedings of the workshop on Computational Aesthetics*. Eurographics Association, 2015, pp. 67–77. [8](#), [12](#), [19](#), [31](#), [36](#), [41](#), [85](#), [91](#), [92](#), [98](#), [99](#), [100](#), [101](#), [103](#), [104](#), [109](#), [111](#), [112](#), [113](#), [114](#), [118](#), [119](#), [121](#), [127](#), [128](#), [141](#), [149](#)
- [8] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 257–266, 2002. [19](#), [119](#), [126](#), [143](#), [145](#)
- [9] R. Mantiuk, K. Myszkowski, and H.-P. Seidel, “A perceptual framework for contrast processing of high dynamic range images,” *ACM Transactions on Applied Perception (TAP)*, vol. 3, no. 3, pp. 286–308, 2006. [19](#), [136](#)
- [10] M. D. Fairchild, *Color appearance models*. John Wiley & Sons, 2013. [27](#), [29](#), [31](#), [112](#), [127](#), [128](#), [143](#)
- [11] S. K. Shevell, *The science of color*. Elsevier, 2003. [27](#)

- [12] C. Ware, “Lightness, brightness, contrast, and constancy,” *Information visualization: perception for design*, 3rd edn. Morgan Kaufmann, Durham, pp. 69–95, 2012. [27](#)
- [13] C. Matters, “Basic color theory,” *Retrieved March*, vol. 27, p. 2015, 2012. [27](#)
- [14] T. Young, “The bakerian lecture: On the theory of light and colours,” *Philosophical transactions of the Royal Society of London*, vol. 92, pp. 12–48, 1802. [28](#)
- [15] O. Noboru and A. R. Robertson, “3.9: Standard and supplementary illuminants, colorimetry,” 2005. [29](#), [31](#)
- [16] G. Wyszecki and W. S. Stiles, *Color science*. Wiley New York, 1982, vol. 8. [29](#), [31](#)
- [17] M. D. Fairchild, “A revision of ciecam97s for practical applications,” *Color Research & Application*, vol. 26, no. 6, pp. 418–427, 2001. [29](#)
- [18] J. Huo, Y. Chang, J. Wang, and X. xia Wei, “Robust automatic white balance algorithm using gray color points in images.” *IEEE Trans. Consumer Electronics*, vol. 52, no. 2, pp. 541–546, 2006. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tce/tce52.html#HuoCWW06> [31](#), [56](#), [113](#)
- [19] N. Limare, J.-L. Lisani, J.-M. Morel, A. B. Petro, and C. Sbert, “Simplest color balance,” *Image Processing On Line*, vol. 1, pp. 297–315, 2011. [31](#)
- [20] O. Frigo, N. Sabater, V. Demoulin, and P. Hellier, “Optimal transportation for example-guided color transfer,” in *Asian Conference on Computer Vision*. Springer, 2014, pp. 655–670. [31](#), [45](#), [56](#), [57](#), [58](#), [61](#), [63](#), [98](#), [128](#)
- [21] J. Kuang, G. M. Johnson, and M. D. Fairchild, “icam06: A refined image appearance model for hdr image rendering,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 5, pp. 406–414, 2007. [33](#), [56](#), [57](#), [116](#), [127](#), [128](#)
- [22] E. Gómez, M. Gomez-Viilegas, and J. Marin, “A multivariate generalization of the power exponential family of distributions,” *Communications in Statistics-Theory and Methods*, vol. 27, no. 3, pp. 589–600, 1998. [35](#), [65](#), [70](#)
- [23] P. J. Davis, “Leonhard euler’s integral: A historical profile of the gamma function: In memoriam: Milton abramowitz,” *The American Mathematical Monthly*, vol. 66, no. 10, pp. 849–869, 1959. [35](#)
- [24] F. Pascal, L. Bombrun, J.-Y. Tourneret, and Y. Berthoumieu, “Parameter estimation for multivariate generalized gaussian distributions,” *Signal Processing, IEEE Transactions on*, vol. 61, no. 23, pp. 5960–5971, 2013. [35](#), [68](#)
- [25] T. S. Cho, C. L. Zitnick, N. Joshi, S. B. Kang, R. Szeliski, and W. T. Freeman, “Image restoration by matching gradient distributions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 4, pp. 683–694, 2012. [35](#), [76](#)

- [26] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on image Processing*, vol. 9, no. 9, pp. 1522–1531, 2000. [35](#)
- [27] D. Cho and T. D. Bui, "Multivariate statistical modeling for image denoising using wavelet transforms," *Signal Processing: Image Communication*, vol. 20, no. 1, pp. 77–89, 2005. [35](#)
- [28] D. Cho, T. D. Bui, and G. Chen, "Image denoising based on wavelet shrinkage using neighbor and level dependency," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 7, no. 03, pp. 299–311, 2009. [35](#), [65](#)
- [29] J. Scharcanski, "A wavelet-based approach for analyzing industrial stochastic textures with applications," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 1, pp. 10–22, 2007. [35](#)
- [30] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance," *Image Processing, IEEE Transactions on*, vol. 11, no. 2, pp. 146–158, 2002. [35](#), [65](#)
- [31] C. Naornita, Y. Berthoumieu, I. Naornita, and A. Isar, "Kullback-leibler distance between complex generalized gaussian distributions," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1850–1854. [35](#)
- [32] G. Verdoolaege and P. Scheunders, "Geodesics on the manifold of multivariate generalized gaussian distributions with an application to multicomponent texture discrimination," *International Journal of Computer Vision*, vol. 95, no. 3, pp. 265–286, 2011. [35](#), [65](#)
- [33] ———, "On the geometry of multivariate generalized gaussian models," *Journal of Mathematical Imaging and Vision*, vol. 43, no. 3, pp. 180–193, 2012. [35](#)
- [34] F. Pitié and A. Kokaram, "The linear monge-kantorovitch linear colour mapping for example-based colour transfer," 2007. [36](#), [41](#), [45](#), [46](#), [65](#), [66](#), [69](#), [74](#), [80](#), [83](#), [85](#), [91](#), [98](#), [99](#), [100](#), [149](#)
- [35] G. N. Watson, *A treatise on the theory of Bessel functions*. Cambridge university press, 1995. [37](#)
- [36] Y. Gong and I. F. Sbalzarini, "Image enhancement by gradient distribution specification," in *Computer Vision-ACCV 2014 Workshops*. Springer, 2014, pp. 47–62. [37](#), [65](#), [76](#)
- [37] T. Eltoft, T. Kim, and T.-W. Lee, "On the multivariate laplace distribution," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 300–303, 2006. [37](#), [65](#)
- [38] A. Al-Saleh and A. El-Zaart, "Unsupervised learning technique for skin images segmentation using a mixture of beta distributions," in *3rd Kuala Lumpur International Conference on Biomedical Engineering 2006*. Springer, 2007, pp. 304–307. [38](#)



- [39] T. Pouli and E. Reinhard, “Progressive histogram reshaping for creative color transfer and tone reproduction,” in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2010, pp. 81–90. [45](#), [99](#), [100](#)
- [40] —, “Progressive color transfer for images of arbitrary dynamic range,” *Computers & Graphics*, vol. 35, no. 1, pp. 67–80, 2011. [45](#)
- [41] X. Xiao and L. Ma, “Gradient-preserving color transfer,” in *Computer Graphics Forum*, vol. 28, no. 7. Wiley Online Library, 2009, pp. 1879–1886. [45](#)
- [42] F. Pitie, A. C. Kokaram, and R. Dahyot, “N-dimensional probability density function transfer and its application to color transfer,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1434–1439. [45](#), [61](#), [99](#), [100](#)
- [43] R. Nguyen, S. Kim, and M. Brown, “Illuminant aware gamut-based color transfer,” in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 319–328. [45](#), [98](#)
- [44] Y. Hwang, J.-Y. Lee, I. So Kweon, and S. Joo Kim, “Color transfer using probabilistic moving least squares,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3342–3349. [45](#), [49](#), [61](#), [98](#), [168](#)
- [45] A. Abadpour and S. Kasaei, “A fast and efficient fuzzy color transfer method,” in *Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International Symposium on*. IEEE, 2004, pp. 491–494. [45](#)
- [46] —, “An efficient pca-based color transfer method,” *Journal of Visual Communication and Image Representation*, vol. 18, no. 1, pp. 15–34, 2007. [45](#)
- [47] C. Villani, *Topics in optimal transportation*. American Mathematical Soc., 2003, no. 58. [46](#), [69](#)
- [48] I. Olkin and F. Pukelsheim, “The distance between two random vectors with given dispersion matrices,” *Linear Algebra and its Applications*, vol. 48, pp. 257–263, 1982. [46](#), [69](#)
- [49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [46](#)
- [50] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [47](#)
- [51] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. [47](#)
- [52] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *arXiv preprint arXiv:1508.06576*, 2015. [47](#), [48](#), [105](#), [168](#)

- [53] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988. [47](#)
- [54] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711. [47](#), [168](#)
- [55] O. Frigo, N. Sabater, J. Delon, and P. Hellier, "Split and match: example-based adaptive patch sampling for unsupervised style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 553–561. [47](#), [105](#), [168](#)
- [56] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," *arXiv preprint arXiv:1703.07511*, 2017. [48](#), [105](#)
- [57] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2479–2486. [48](#)
- [58] V. Wisslar, *Illuminated Pixels: The Why, What, and How of Digital Lighting*. Cengage Learning, 2013. [52](#)
- [59] M. Stone, *A field guide to digital color*. CRC Press, 2013. [52](#)
- [60] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 123–137, 2007. [55](#), [58](#), [59](#), [82](#), [83](#), [113](#), [120](#), [121](#)
- [61] I. Olkin and S. Rachev, "Maximum submatrix traces for positive definite matrices," *SIAM journal on matrix analysis and applications*, vol. 14, no. 2, pp. 390–397, 1993. [55](#)
- [62] D. Dowson and B. Landau, "The frechet distance between multivariate normal distributions," *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982. [55](#)
- [63] L. C. Evans, "Partial differential equations and monge-kantorovich mass transfer," *Current developments in mathematics*, vol. 1999, pp. 65–126, 1997. [55](#), [149](#)
- [64] M.-S. Yang, "A survey of fuzzy clustering," *Mathematical and Computer modelling*, vol. 18, no. 11, pp. 1–16, 1993. [55](#)
- [65] R. Nock and F. Nielsen, "On weighting clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 8, pp. 1223–1235, 2006. [55](#)
- [66] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000. [56](#)
- [67] B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot, "Automatic texture guided color transfer and colorization," in *Expressive 2016*, 2016. [59](#), [63](#)

- [68] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. [61](#), [98](#), [118](#), [131](#)
- [69] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics*, pp. 401–406, 1946. [61](#), [118](#)
- [70] F. J. Aherne, N. A. Thacker, and P. I. Rockett, "The bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998. [61](#), [98](#)
- [71] F. Wu, W. Dong, Y. Kong, X. Mei, J.-C. Paul, and X. Zhang, "Content-based colour transfer," in *Computer Graphics Forum*, vol. 32, no. 1. Wiley Online Library, 2013, pp. 190–203. [63](#)
- [72] S. Asmussen, *Applied probability and queues*. Springer Science & Business Media, 2008, vol. 51. [65](#)
- [73] D. Kelker, "Distribution theory of spherical distributions and a location-scale parameter generalization," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 419–430, 1970. [65](#), [70](#)
- [74] G. Verdoolaege, S. De Backer, and P. Scheunders, "Multiscale colour texture retrieval using the geodesic distance between multivariate generalized gaussian models," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 169–172. [65](#)
- [75] M. S. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet, "Image and video segmentation by combining unsupervised generalized gaussian mixture modeling and feature selection," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 10, pp. 1373–1377, 2010. [65](#)
- [76] I. W. Selesnick *et al.*, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *Signal Processing, IEEE Transactions on*, vol. 50, no. 11, pp. 2744–2756, 2002. [65](#)
- [77] Y. Xiang, B. Zou, and H. Li, "Selective color transfer with multi-source images," *Pattern Recognition Letters*, vol. 30, no. 7, pp. 682–689, 2009. [66](#)
- [78] V. K. Nath and A. Mahanta, "Image denoising based on laplace distribution with local parameters in lapped transform domain," in *Signal Processing and Multimedia Applications (SIGMAP), 2011 Proceedings of the International Conference on*. IEEE, 2011, pp. 1–6. [66](#)
- [79] H. Rabbani and M. Vafadust, "Image/video denoising based on a mixture of laplace distributions with local parameters in multidimensional complex wavelet domain," *Signal Processing*, vol. 88, no. 1, pp. 158–173, 2008. [66](#)
- [80] H. J. Park and T. W. Lee, "Modeling nonlinear dependencies in natural images using mixture of laplacian distribution," in *Advances in neural information processing systems*, 2004, pp. 1041–1048. [66](#)

- 
- [81] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Corporation, 1964, vol. 55. 68
- [82] L. Bombrun, F. Pascal, J.-Y. Tournet, and Y. Berthoumieu, “Performance of the maximum likelihood estimators for the parameters of multivariate generalized gaussian distributions,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3525–3528. 68
- [83] K. Yao, “A representation theorem and its applications to spherically-invariant random processes,” *IEEE Transactions on Information Theory*, vol. 19, no. 5, pp. 600–608, 1973. 70
- [84] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, “Complex elliptically symmetric distributions: Survey, new results and applications,” *IEEE Transactions on signal processing*, vol. 60, no. 11, pp. 5597–5625, 2012. 70
- [85] K. Krishnamoorthy and T. Mathew, *Statistical tolerance regions: theory, applications, and computation*. John Wiley & Sons, 2009, vol. 744. 71, 173
- [86] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” in *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 313–318. 75
- [87] M. W. Tao, M. K. Johnson, and S. Paris, “Error-tolerant image compositing,” *International journal of computer vision*, vol. 103, no. 2, pp. 178–189, 2013. 75
- [88] P. Bhat, B. Curless, M. Cohen, and C. L. Zitnick, “Fourier analysis of the 2d screened poisson equation for gradient domain problems,” in *Computer Vision—ECCV 2008*. Springer, 2008, pp. 114–128. 75, 76, 158, 159
- [89] S. Bae, S. Paris, and F. Durand, “Two-scale tone management for photographic look,” in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 637–645. 75
- [90] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 249–256. 75
- [91] F. Deng, S. J. Kim, Y.-W. Tai, and M. S. Brown, “Color-aware regularization for gradient domain image manipulation,” in *Computer Vision—ACCV 2012*. Springer, 2013, pp. 392–405. 75, 76, 159
- [92] J.-M. Morel, A.-B. Petro, and C. Sbert, “Screened poisson equation for image contrast enhancement,” *Image Processing On Line*, vol. 4, pp. 16–29, 2014. 75, 158
- [93] W. Xu and J. Mulligan, “Performance evaluation of color correction approaches for automatic multi-view image and video stitching,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 263–270. 82
- [94] Z. Ma, “Non-gaussian statistical models and their applications,” Ph.D. dissertation, 2011. 83

- [95] E. Paulson, “An approximate normalization of the analysis of variance distribution,” *The Annals of Mathematical Statistics*, vol. 13, no. 2, pp. 233–235, 1942. [88](#)
- [96] T. Ashby, “A modification to paulson’s approximation to the variance ratio distribution,” *The Computer Journal*, vol. 11, no. 2, pp. 209–210, 1968. [88](#), [89](#)
- [97] E. Fieller, “The distribution of the index in a normal bivariate population,” *Biometrika*, pp. 428–440, 1932. [88](#)
- [98] D. M. Hawkins and R. Wixley, “A note on the transformation of chi-squared variables to normality,” *The American Statistician*, vol. 40, no. 4, pp. 296–298, 1986. [88](#), [90](#), [173](#)
- [99] O. Le Meur and T. Baccino, “Methods for comparing scanpaths and saliency maps: strengths and weaknesses,” *Behavior research methods*, vol. 45, no. 1, pp. 251–266, 2013. [101](#)
- [100] O. Le Meur and Z. Liu, “Saliency aggregation: Does unity make strength?” in *Asian Conference on Computer Vision*. Springer, 2014, pp. 18–32. [101](#)
- [101] J. Harel, C. Koch, P. Perona *et al.*, “Graph-based visual saliency,” in *NIPS*, vol. 1, no. 2, 2006, p. 5. [101](#), [160](#), [161](#)
- [102] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit, “Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis,” *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 642–658, 2013. [101](#)
- [103] M. D. Fairchild and G. M. Johnson, “icam framework for image appearance, differences, and quality,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 126–138, 2004. [101](#), [127](#)
- [104] S. F. Crone and S. Finlay, “Instance sampling in credit scoring: An empirical study of sample size and balancing,” *International Journal of Forecasting*, vol. 28, no. 1, pp. 224–238, 2012. [102](#)
- [105] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002. [102](#)
- [106] J.-P. Vert, K. Tsuda, and B. Schölkopf, “A primer on kernel methods,” *Kernel Methods in Computational Biology*, pp. 35–70, 2004. [102](#)
- [107] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001. [102](#), [104](#)
- [108] R. Ramirez Orozco, C. Loscos, I. Martin, and A. Artusi, *Multiview HDR video sequence generation*. Academic Press, Elsevier, apr 2016, pp. 121–138. [110](#)

- 
- [109] E. Reinhard and T. Pouli, "Colour spaces for colour transfer," in *Computational Color Imaging*. Springer, 2011, pp. 1–15. [112](#)
- [110] M. D. Fairchild and P.-H. Chen, "Brightness, lightness, and specifying color in high-dynamic-range scenes and images," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, pp. 78 670O–78 670O. [114](#)
- [111] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 267–276. [115](#), [116](#), [118](#), [119](#), [120](#)
- [112] R. Boitard, R. Cozot, D. Thoreau, and K. Bouatouch, "Zonal brightness coherency for video tone mapping," *Signal Processing: Image Communication*, vol. 29, no. 2, pp. 229–246, 2014. [115](#), [116](#)
- [113] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 40. [117](#), [133](#)
- [114] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *ACM SIGGRAPH 2008 classes*. ACM, 2008, p. 31. [124](#), [126](#), [130](#)
- [115] S. Mann and R. Picard, "On being "undigital" with digital cameras: Extending dynamic range by combining differently exposed pictures," in *Proceedings of Society for Imaging Science and Technology's 48th Annual Conference*, 1995. [124](#), [126](#)
- [116] O. Gallo, N. Gelfand, W.-C. Chen, M. Tico, and K. Pulli, "Artifact-free high dynamic range imaging," in *Computational Photography (ICCP), 2009 IEEE International Conference on*. IEEE, 2009, pp. 1–7. [124](#)
- [117] D. Sidibé, W. Puech, and O. Strauss, "Ghost detection and removal in high dynamic range images," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 2240–2244. [124](#)
- [118] M. Granados, K. I. Kim, J. Tompkin, and C. Theobalt, "Automatic noise modeling for ghost-free hdr reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 201, 2013. [124](#)
- [119] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based HDR reconstruction of dynamic scenes," *ACM Trans. Graph.*, vol. 31, no. 6, p. 203, 2012. [124](#)
- [120] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A versatile hdr video production system," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4, p. 41, 2011. [124](#)
- [121] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1. IEEE, 2000, pp. 472–479. [124](#)



- [122] S. K. Nayar and S. G. Narasimhan, "Assorted pixels: Multi-sampled imaging with structural models," in *Computer Vision—ECCV 2002*. Springer, 2002, pp. 636–652. [124](#)
- [123] C. Aguerrebere, A. Almansa, Y. Gousseau, J. Delon, and P. Muse, "Single shot high dynamic range imaging using piecewise linear estimators," in *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–10. [125](#)
- [124] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia, "Convolutional sparse coding for high dynamic range imaging," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley-Blackwell, 2016. [125](#)
- [125] Z. Li and J. Zheng, "Single image brightening via exposure fusion," *ICASSP*, 2016. [125](#), [136](#), [137](#), [169](#)
- [126] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion: A simple and practical alternative to high dynamic range photography," in *Computer Graphics Forum*, vol. 28, no. 1. Wiley Online Library, 2009, pp. 161–171. [125](#), [135](#), [136](#)
- [127] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," in *ACM Transactions on Graphics (Proceedings of Siggraph Conference)*, vol. 23. ACM Press, 2004. [Online]. Available: <http://maverick.inria.fr/Publications/2004/ED04> [125](#), [126](#), [127](#), [135](#), [136](#), [143](#), [145](#)
- [128] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," in *ACM transactions on graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 664–672. [125](#), [127](#), [143](#), [145](#)
- [129] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 1. [125](#)
- [130] Y. Gryaditskaya, T. Pouli, E. Reinhard, K. Myszkowski, and H.-P. Seidel, "Motion aware exposure bracketing for hdr video," in *Computer Graphics Forum*, vol. 34, no. 4. Wiley Online Library, 2015, pp. 119–130. [125](#)
- [131] N. K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D. B. Goldman, and P. Sen, "Patch-based high dynamic range video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 202–1, 2013. [125](#)
- [132] R. Matsuoka, T. Baba, M. Okuda, and K. Shirai, "High dynamic range image acquisition using flash image," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 1612–1616. [125](#)
- [133] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European conference on computer vision*. Springer, 2010, pp. 1–14. [126](#), [143](#), [145](#), [146](#)
- [134] C. Bist, R. Cozot, G. Madec, and X. Ducloux, "Style aware tone expansion for hdr displays," *Graphics Interface (GI)*, 2016. [127](#)

- 
- [135] T. Matsuo, N. Fukushima, and Y. Ishibashi, "Weighted joint bilateral filter with slope depth compensation filter for depth map refinement." in *VISAPP (2)*, 2013, pp. 300–309. [143](#), [145](#)
- [136] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European Conference on Computer Vision*. Springer, 2014, pp. 815–830. [145](#)
- [137] I. Omer and M. Werman, "Color lines: Image specific color representation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–II. [146](#)
- [138] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2008. [146](#)
- [139] W.-C. Tu, C.-L. Tsai, and S.-Y. Chien, "Collaborative noise reduction using color-line model," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2465–2469. [146](#)
- [140] A. Zomet and S. Peleg, "Multi-sensor super-resolution," in *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*. IEEE, 2002, pp. 27–31. [146](#)
- [141] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering." *IEEE Transactions on Image processing*, vol. 24, no. 1, pp. 120–129, 2015. [147](#)
- [142] X. Zhang, T. Sim, and X. Miao, "Enhancing photographs with near infra-red images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. [159](#)
- [143] H. Seo and P. Milanfar, "Computational photography using a pair of flash/no-flash images by iterative guided filtering," in *IEEE International Conference on Computer Vision (ICCV)*, 2011. [159](#)
- [144] H.-J. Seo and P. Milanfar, "Robust flash denoising/deblurring by iterative guided filtering," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 1–19, 2012. [159](#), [160](#)
- [145] S. Zhuo, D. Guo, and T. Sim, "Robust flash deblurring," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2440–2447. [159](#), [160](#)
- [146] W. Li, J. Zhang, and Q.-h. Dai, "Robust blind motion deblurring using near-infrared flash image," *Journal of Visual Communication and Image Representation*, vol. 24, no. 8, pp. 1394–1413, 2013. [159](#)
- [147] S. Ssstrunk, C. Fredembach, and D. Tamburrino, "Automatic skin enhancement with visible and near-infrared image fusion," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1693–1696. [161](#)

- [148] L. Liang, L. Jin, and X. Li, “Facial skin beautification using adaptive region-aware masks,” *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2600–2612, 2014. [162](#)
- [149] C. Fredembach, N. Barbuscia, and S. Süsstrunk, “Combining visible and near-infrared images for realistic skin smoothing,” in *Color and Imaging Conference*, vol. 2009, no. 1. Society for Imaging Science and Technology, 2009, pp. 242–247. [161](#)
- [150] V. Gaikar, “Make pictures perfect with pictreat,” 2009. [162](#)
- [151] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, “Palette-based photo recoloring,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 139, 2015. [166](#)
- [152] C. H. Nguyen, T. Ritschel, and H.-P. Seidel, “Data-driven color manifolds,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, p. 20, 2015. [167](#)
- [153] S. Kotz, N. Balakrishnan, and N. L. Johnson, *Continuous multivariate distributions, models and applications*. John Wiley & Sons, 2004. [167](#)
- [154] N. Bouguila, D. Ziou, and J. Vaillancourt, “Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application,” *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1533–1543, 2004. [167](#)
- [155] T. Lumley, “Kendall’s advanced theory of statistics. volume 2a: classical inference and the linear model. alan stuart, keith ord and steven arnold, arnold, london, 1998, no. of pages: xiv+ 885. price:£ 85.00. isbn 0-340-66230-1,” *Statistics in Medicine*, vol. 19, no. 22, pp. 3139–3140, 2000. [167](#)
- [156] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, “Data-driven hallucination of different times of day from a single outdoor photo,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 200, 2013. [169](#)
- [157] S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High dynamic range video,” *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 319–325, 2003. [170](#)
- [158] S. Mangiat and J. Gibson, “Spatially adaptive filtering for registration artifact removal in hdr video,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1317–1320. [170](#)
- [159] N. K. Kalantari, E. Shechtman, C. Barnes, S. Darabi, D. B. Goldman, and P. Sen, “Patch-based High Dynamic Range Video,” *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH Asia 2013)*, vol. 32, no. 6, 2013. [170](#)
- [160] M. Hudon, A. Gruson, P. Kerbiriou, R. Cozot, and K. Bouatouch, “Shape and reflectance from rgb-d images using time sequential illumination,” in *Intenational Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 64, 2016, pp. 65–66. [170](#)

- [161] D. Khoshnevisan, *Probability*, ser. Graduate studies in mathematics. American Mathematical Society, 2007. [Online]. Available: <https://books.google.fr/books?id=BgYPCgAAQBAJ> 171
- [162] E. B. Wilson and M. M. Hilferty, “The distribution of chi-square,” *proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 12, p. 684, 1931. 173